Sheffield Hallam University

Gaseous fuel mixture parameter prediction in a steel works using minimalist neural networks.

PITTMAN, Manfred.

Available from the Sheffield Hallam University Research Archive (SHURA) at:

http://shura.shu.ac.uk/20233/

A Sheffield Hallam University thesis

This thesis is protected by copyright which belongs to the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Please visit http://shura.shu.ac.uk/20233/ and http://shura.shu.ac.uk/information.html for further details about copyright and re-use permissions.

Learning and IT Services
Adsetts Centre City Campus
Sheffield S1 1WB

101 853 253 6

of issue

REFERENCE

ProQuest Number: 10700878

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10700878

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Gaseous fuel mixture parameter prediction in a steelworks using minimalist neural networks

Manfred Pittman

A thesis submitted in partial fulfillment of the requirements of Sheffield Hallam University for the degree of Doctor of Philosophy

March 2005

Gaseous fuel mixture parameter prediction in a steelworks using minimalist neural networks

A thesis submitted in partial fulfillment of the requirements of Sheffield Hallam University for the degree of Doctor of Philosophy

by Manfred Pittman

Abstract

This thesis reports a developmental programme of work undertaken for a number of installations at the Scunthorpe site of Corus plc (formerly British Steel plc).

The thesis commences with an introduction and outline of the problems, together with the objectives of the research This is followed by a survey of literature and work by other groups, and an introduction to relevant aspects of neural networks. Chapter 4 details the experimental background and procedures, whilst chapters 5 – 9 are concerned with analysis of data and results relating to specific installations. Finally, chapter 10 presents an overall review of the results, with conclusions and suggestions for further work. Tables of typical results are listed in the appendices.

The central theme concerns the optimisation of the combustion of gas mixes produced by on-site processes through parameter prediction, the constituents of the mixtures included blast furnace gas (BFG), coke oven gas (COG), and basic oxygen steelmaking gas (BOS). The main parameters under investigation were calorific value (CV), air/fuel ratio, and specific gravity. Finally, a secondary investigation was conducted into predicting oxygen content in flue gases with a view to reducing recuperator corrosion.

Data from three different systems was considered: a power station, a coke oven gas plant and a section mill. All data sets were subject to aliasing both as a result of the slowness of the mass spectrometer measuring devices with respect to gas content fluctuations, and as a result of the relatively long sampling interval employed by the main archiving system. The sets from the section mill were particularly prone to abrupt and extreme variations.

Neural network solutions based on function approximation were proposed and developed. There was a specific requirement that any solution be compatible with unsophisticated low-budget hardware. Hence there were major constraints on network size and complexity. A linear time series based network was found to perform more efficiently in the data supplied rather than the more conventional non-linear counterpart.

The proposed networks indicated potential gains in accuracy in excess of 50% over a second-order least squares-based method proposed by the collaborating organisation.

At the time of writing it is understood that no other similar systems have been investigated in this manner, let alone resulting in a successful minimalist neural network solution. Hence the contribution to knowledge is that it is possible to accurately predict the above parameters with a minimalist linear network, trained with data subjected to varying degrees of aliasing.

Acknowledgements

The author would like to thank his director of study Dr WA Barraclough and second supervisors Dr MJ Denman and Dr K Dutton for their valued advice, support and patient understanding over the extended time required to complete this work. There was no formal agreement between Sheffield Hallam University and Corus and hence reliance was placed on the goodwill of several staff who kindly supplied data and advice when time and circumstances permitted; my thanks also to them. A research project is nothing without effective sources of information so there is a debt of gratitude to several members of the Sheffield Hallam library staff who assisted me in many ways; particularly acknowledged are the contributions of Roger Hines (Science and Engineering Team), Laura Tolley (Document Supply Services), and their respective staffs. Finally, my profound thanks to my family and friends for their support, patience and understanding; I hope to have more time for them all in the not too distant future!

Manfred Pittman, February 2005

Contents

List of figures

Tables in main text

(Tables of typical results are given in the Appendices)

Chapter 8. *HSM Phase 2*

Chapter 9. *Recuperator Corrosion*

1.Introduction and objectives

This thesis describes investigations into the development of neural network-based predictors with the purpose of forecasting parameters in gas mixes of varying complexity in several different installations at Corus' (formerly British Steel's) Scunthorpe site. These parameters in the main consisted of calorific value (CV), specific gravity (s.g.), and air/fuel ratio (a/f), with some work in the later stages involving Wobbe indices, and percentage oxygen content in exhaust flue gases. (This thesis follows the particular industry convention of using 'specific gravity' rather than the more contemporary 'relative density'.) The Wobbe number or index is used as a means of comparing the heat inputs to a burner of different gases at fixed pressure, and is defined as

$$W = \frac{CV}{\sqrt{s.g.}} MJ/kg \qquad \text{——————} (1.1)$$

where CV = calorific value, s.g. = specific gravity

Discussions with Corus staff had revealed a requirement for parameter prediction in order to assist with the optimisation of combustion systems on economic and environmental grounds. On-site processes produced a variety of gases as bye-products, such as blast furnace gas (BFG), coke oven gas (COG), and basic oxygen steel-making gas (BOS), which are then mixed and distributed throughout the site for use as fuels. (Appendix A1 tabulates typical properties.) The cost of these compared to that of importing fuels from external sources (e.g. heavy fuel oil or natural gas) was described as "negligible". Further, the author was informed that forthcoming environmental legislation would restrict the practice of 'flaring off' unused fuel gas; it was essential that as much energy as possible be recovered from on-site gas mixes.

Corus staff also stated that these mixes could vary widely in content, often abruptly, and that there were significant transportation time-lags within the distribution system, the effects of these being exacerbated by the relatively slow operation of the associated valve-trains (up to 20 seconds). In addition, due to budget constraints any potential solution would have to be compatible with low-specification hardware.

The author had been involved in some earlier investigations into the feasibility of developing an expert system to optimise combustion processes. This had involved both Genetic

Algorithms (GAs) and neural networks. Whilst GAs were capable of highly accurate solutions, those solutions were specific to a given dataset. In contrast, neural networks [e.g. Hayk99, p205 et seq] are capable of generalisation and thus have an inherent ability to cope with previously unseen data. Hence a neural network approach was adopted. However potential network size and sophistication were constrained by the above limitations on hardware specification. Minimalist, or parsimonious, solutions were sought. Initially some non-linear networks were developed on the understanding that redundant PCs might be available; later in the project information was received that only PLCs were available and non-linear work ceased. (Nonetheless, the non-linear work is reported in the earlier chapters.) Mass spectrometer datasets from the fuel distributions systems of the following departments were investigated:

Central Power Station – CPS (chapter 5)
Heavy Section Mill – HSM (chapters 6 and 8)
Dawes Lane Coke Ovens (chapter 7)

In addition exhaust flue data (percentage oxygen content) from the HSM was also examined in respect of a recuperator corrosion problem (chapter 9).

Data from the mass spectrometers was received in the form of spreadsheets largely from the main archiving system (sampled at 1-minute intervals) with some data recorded directly at the mass spectrometer outputs (sampled at 30- or 23-second intervals depending on mass spectrometer type).

No indication of acceptable accuracy levels was provided but it was learnt that a second-order least squares-based predictor (based on the previous 5 values) had been investigated by Corus and found to give acceptable results. Such a predictor was developed in the course of this research and used to provide benchmarking for the proposed neural network solutions.

1.1 Objectives

- To provide minimalist neural network predictors capable of being implement on unsophisticated low-specification hardware, predicting specified parameters up to two sampling intervals ahead.
- These parameters consisted of CV, Wobbe index, s.g. and air/fuel ratio, from gas mixes of varying complexity, and with constituent gases including COG, BFG and BOS.

- Predictor performance was to attain greater accuracy than that of the benchmark least squares implementation, and to demonstrate greater resilience when operating with aliased data, e.g. data from the main archiving system as opposed to that obtained directly from mass spectrometer outputs.
- A secondary investigation would attempt to apply the method to predicting oxygen content in exhaust flue gas in respect of a recuperator corrosion problem.

2.0 Survey of Literature and Work by Other Groups.

(References to literature relevant to general neural network topics are contained in the appropriate sections in subsequent chapters.)

2.1 QUB (Queen's University Belfast)

In the UK many of the papers published concerning modelling, and control and optimisation, of power stations emanate from Queen's University, Belfast. (The Control of Power Systems Group led by Professor B.W.Hogg.)

This work commenced in the 1980s and much of it is centred around the Ballylumford Power Station in collaboration with Northern Ireland Electric (NIE). (Ballylumford is a conventional oil–fired electricity generating station supplying the national grid.)

The bulk of the research is based on classical differential/algebraic equation techniques. It ranges from boiler identification [Chawdry89] to object–oriented modelling techniques (using the C++ Programming Language)[Lu94].

These papers are somewhat descriptive in nature and lacking in detail to be capable of any immediate contribution to this research. It is assumed that there is a measure of commercial confidentiality involved.

More recently, the group's work has begun to embrace soft computing methods with the design of a proposed Adaptive Fuzzy Logic controller for AVR (Automatic Voltage Regulation) [Lown97].

Nothing in their published work suggests the study of mixed–fuel systems, nor prediction of fuel parameters.

2.2 University of Strathclyde

This is a group from that university's Industrial Control Centre led by Professor Michael J. Grimble, and working in conjunction with Scottish Power and John Brown Engineering. The text consulted was "*Modelling and Simulation of Power Generation Plants*". [Ordys94]

Much of the early investigative work in this research involved verification of the modelling presented in the book, using Simulink. Much of the data presented proved to have ambiguities, or had been omitted, and work ceased.

Again there appears to be no involvement with mixed–fuel systems, nor the prediction of fuel parameters.

2.3 K.F. Reinschmidt

Dr K.Reinschmidt (Senior Vice President of Stone & Webster Engineering Group) has published a number of papers concerning the involvement of neural networks in power generation.

Two of the earlier papers read [Rein91, Rein94] concern the modelling and control of power plants. However he states that the data used was obtained from an earlier work describing a "utility once–through boiler" (a type that does not feature in this research).

The second two papers on "*Short term electrical load forecasting*" [Rein95a, Rein95b] , employing neural networks, genetic algorithms, and knowledge–based systems, are of particular relevance and interest to this project.

In these, Reinschmidt describes a method for forecasting demand at an electricity generating station, on an hourly basis. He utilises such input data as temperature, humidity and other meteorological data, together with other factors such as holidays.

There are two types of neural network employed - linear and non-linear. The former is a single-layer network with a linear activation function employing a 'time series' model, and the latter a multilayer network with non-linear activation functions.

The following time series is utilised in both the GA-based and linear neural network approaches:

$$z_t = a_1 z_{t-1} + a_2 z_{t-2} + \ldots + a_p z_{t-p} + c_0 x_t + c_1 x_{t-1} + \ldots + c_p x_{t-p} + u_t \quad \text{—— (2.1)}$$

where z_t = electrical load at time t

x_t = independent variable

u_t = random disturbance

a_i, c_i are coefficients

Reinschmidt states in his April 1995 paper [Rein95a] that

> "Linear neural networks can successfully learn the coefficients ... from historical load data and the independent variables. Both ... Widrow-Hoff ... and back propagation have been used ... both give equivalent results, but Widrow-Hoff is faster in training."

However, he also states that genetic algorithms were also used to establish the parameters in the linear time series equation, commenting that

> "The genetic algorithm often gives forecasts as accurate as those obtained from neural networks ...However, the results ... are more variable than those from the neural network method."

The non-linear network uses the sigmoid activation function and consists "typically" of one hidden layer although "in some cases two", and has no bias terms in the output layer. The standard back-propagation training method is used. Of great interest to this research, especially the concept and development of a parsimonious network, he comments

> "Unlike the linear time series model, in which there is one fitted coefficient for each lagged variable, in the nonlinear neural network forecaster the selection of lagged input variables is independent of the number of fitted coefficients, the network weights, the number of which is determined by the number of layers and the number of hidden units. Also, in linear regression models, if an input is extraneous, then its regression coefficient is zero (or more properly, is not significantly different from zero ...). However, in nonlinear neural networks this is not necessarily true; an input variable may be unimportant but still have large weights; the effects of these weights cancel somewhere downstream.
>
> Therefore, in conventional backpropagation for nonlinear neural networks, there is no automatic elimination of extraneous input nodes or hidden nodes. However in practical forecasting it is necessary to achieve a parsimonious model, one which is neither too simple nor too complex for the problem at hand. If the neural network is chosen to be too small (to have too few inputs or hidden units), then it will not be flexible enough to capture the dynamics of the ... system; this is known as underfitting. Conversely, if the neural network is too large, then it can fit not only the underlying signal but also the noise in the training set; this is known as overfitting. Overfitted models may show low error rates on the training set but do not generalise; they may then have high error rates in actual prediction.
>
> The nonlinear model can yield greater accuracy than the linear formulation, but takes longer to train. Large nonlinear neural networks are also prone to overfitting. Forecasting requires parsimonious models capable of generalisation. The size of the neural network can be reduced by examining the correlation coefficients, or by using genetic algorithms to select the optimum set of input variables. The linear model is a satisfactory approximation to the nonlinear model for the purpose of selecting the input terms.
>
> Large ... neural networks trained using backpropagation are notoriously time-consuming, and a number of methods to reduce training time have been evaluated. One method that had been found to yield orders of magnitude reductions in training time replaces the steepest descent search by techniques that modify the network weights using a least-squares approach; the computations in each step are greater but the number of training iterations is greatly reduced. Reductions in training time are desirable not only to reduce computation costs, but to allow more alternative input variables to be investigated, and hence to optimise forecast accuracy.

He concludes, that by using all three methods and incorporating them into a rule-based expert system, that

> "it is possible to produce composite forecasts that are more robust and more accurate than any single method."

Whilst this research concerns parameter forecasting over much shorter periods than Reinschmidt's hourly load forecasting, his discussion provides in the absence of other criteria, some pointers/indications for potential network design and evaluation. In particular his use of MAPE (mean absolute percentage error) as an accuracy measurement. However little in the way of advice on number of cells and layers is offered.

2.4 Boccaletti et al

The paper "*An application of neural network in combustion process evaluations*" [Bocc99] refers to a power plant but does not indicate its purpose or fuel used. However it is one of the few papers in this field that provides details of neural network architecture and training methods, together with a discussion of the justification for the employment of neural networks and a discussion of results and accuracy, and as such is of interest to this research.

The neural computing approach was adopted because of the complexity of the non-linear equations utilised in conventional methods and the consequent demands on computing time and resources.

A feed-forward network consisting of one hidden layer with log sigmoid activation functions and trained by back propagation was used to obtain highly accurate results in predicting combustion output conditions.

Of the network, he comments

> "Choosing an appropriate number of hidden neurons is extremely important. Using too few will starve the network. Using too many hidden neurons will increase the training time, perhaps so much that it becomes very difficult to train the network adequately in a reasonable period of time. The present study has been started with four hidden neurons and subsequently increased in a step of one. (sic)"

and whilst discussing the choice of activation function, he adds

> " .... In most cases, it has been found that the exact shape of the activation function has little effect on the ultimate power of the network, though it can have a significant impact on training speed."

The chosen data set sizes are listed as:

Sample size	*Case #1*	*Case #2*
Total data	3982	3982
Training	2982 (75%)**	2582 (56%)
Validation	600 (15%)	1000 (25%)
Cross-validation *	400 (10%)	400 (10%)

*Termed the verification set in this research.
** Percentage of the available data.

Table 2.1 relative sizes of datasets (see text)

The results for his optimised networks are shown below :

Parameters	*Case #1*	*Case #2*
Inputs	13	13
Hidden neurons	45	22
Output neurons	6	6
Learning rate	0.7	0.7
Training sample size	2982	2582
Epoch size	18,000	20,000
Training error	0.00993	0.002516
Testing error	0.00990	0.002544

Table 2.2 results (see text)

It should be noted that Case #2 with the larger validation set required fewer hidden neurons and provided a near three-fold increase in accuracy. However, Boccaletti draws attention to the fact that in Case #2 one of the output parameters was changed for another whose

> " ... range of variability ....is smaller ... This reduces the network complexity, and consequently the number of hidden layer neurons and the training time."

(It is this author's view that a further factor might be that a larger validation set may have increased training efficiency.)

Boccaletti concludes

> "This work has demonstrated that it is possible to describe by means of neural networks the connections among quantities involved in physical, chemical and thermodynamic processes."

and adds that

> "1. It is not only the number of input and output variables which affect the learning, but also the relationships between the input and output variables play an important role.
> 2. The number of hidden neurons depends not only on the number of input and output neurons, but much more on the relationship between the input and output. Higher the the complexity of the problem, greater the number of hidden neurons."

It should be noted that the data described in this paper was contrived : random number routines were used to generate artificial data within known ranges. Further, the sizes of validation and verification sets seem small compared to that of the training set.

Nonetheless, the comments on training and structure are of interest.

2.5 Other publications

Cui and Shin [Cui92] apply a neural network to temperature control of a 'once-through' boiler using a multi-layer perceptron network, with two hidden layers, trained by back-propagation. In contrast to other texts – e.g. Boccaletti above – they assert that

> "It is proved that a four-layer (with two hidden) perceptron can be used to approximate any continuous function with the desired accuracy."

The number of hidden cells

> " ... depends on the controlled plant under consideration and is selected experimentally."

and eventually conclude for that system, that increasing the number of hidden neurons beyond a certain point

> " ... does not improve system performance. But adding more nodes will improve the system's reliability."

Khalil and Omatu [Khalil92] describe the use of a neural network in the control of a non-linear heating process. The learning scheme is a two-phase process involving training a network as a plant emulator, followed by the training of a second network as a controller. Essentially this is an instance of function approximation using a three-layer network with one hidden layer containing eight neurons with sigmoid activation functions trained by a variation of the generalised delta rule. It is also stated that the number of hidden neurons is determined empirically.

Other work relates to using AI/Soft–Computing for load prediction and performance enhancement. Gibson [Gib96] describes a commercial system (for conventional power plants) and is therefore somewhat sparse in detail, whilst Krost [Krost96, Krost98] deals with a photo–voltaic cell installation. Sanz [Sanz95]details a "multi–fuel" system which is in fact a hydro–electric plant with a diesel back–up generator, rather than a "mixed fuel" system such as the one under investigation in this project.

In conclusion, none of the publications found to date describe any mixed (fossil) fuel systems, let alone anything approaching the diversity of that at the BS (British Steel) Scunthorpe site.

3.0 Neural Networks

3.1 Intoduction

A neural network is

> "....an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been developed as generalisations of mathematical models of human cognition or neural biology, based on the assumptions that:
>
> 1. Information processing occurs at many simple elements called neurons.
> 2. Signals are passed between neurons over connection links.
> 3. Each connection has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
> 4. Each neuron applies an activation function (usually non-linear) to its net input (sum of weighted inputs) to determine its output signal. "
>
> [Faus94, p3]

Neural networks, sometimes referred to as artificial neural networks (ANNs), are therefore a paradigm based on biological nervous systems and their nerve cells (neurons). Such a biological nerve cell may be shown as

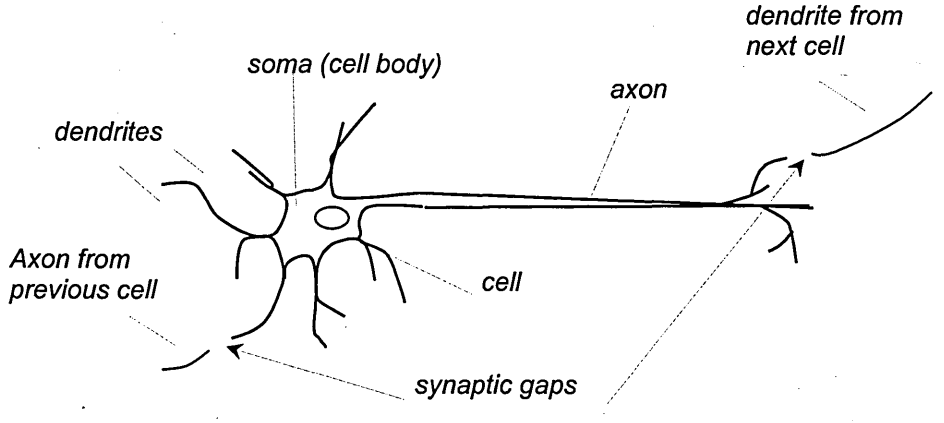


fig. 3.1 biological nerve cell, or neuron

The dendrites form inputs (receptors) to the cell, and the axon the output to the next cell(s). Axons and dendrites are connected by *synapses*. Electrical signals from one cell to another are transmitted across the *synaptic gap* by chemical action, which modifies the signal by scaling it. This scaling can be viewed as being analogous to the weighting in artificial neural networks. (See below.) The soma or cell body sums the incoming signals and when a particular threshold has been reached a signal is sent out along the axon.

Figure 3.2 (below) is a functional representation, which may be expressed in mathematical terms, with reference to the input values *i* and their weightings *w*, as follows:

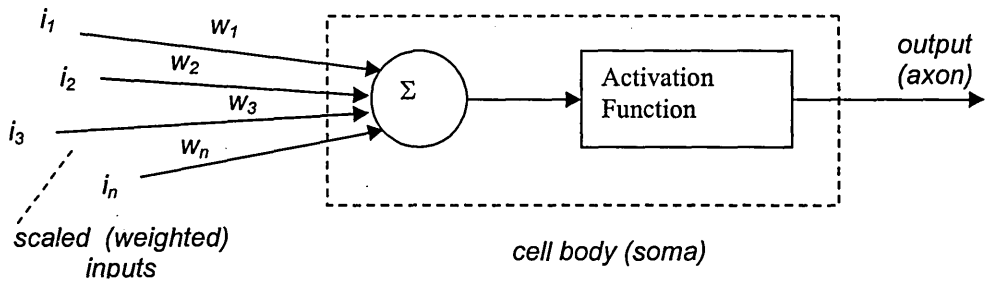


fig. 3.2 functional representation of neuron

$$output = f_{activation}\left(\sum_{i=1}^{n} i_i w_i \right) \quad \text{—— (3.1)}$$

Which in turn may also be expressed in vector form as

$$output = f_{activation}\left(iw + b \right) \quad \text{—— (3.2)}$$

where *b* is an optional bias term (see below) and functions as an additional input of constant value 1 and of weighting *b*.

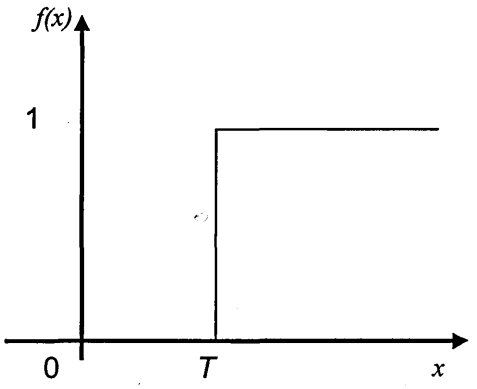
The values of the weights (including the bias *b*) may be determined either by inspection or training.

In the course of the development of artificial neural networks technology, several forms of activation function emerged; those relevant to this research are described below.

One of the earliest practical examples was the thresholding function (fig. 3.3 below) based on that proposed by McCulloch and Pitts in their 1943 paper which according to Hagan et al.

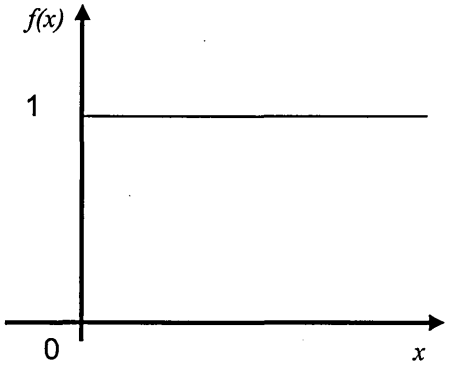
> "showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function." [Hagan95, p 1-3]

The threshold value (*T*) may be adjusted to suit the application. However, use of a bias term will have the same effect of shifting the plot along the x-axis, and thus *T* can be set to zero, as in fig. 3.4, which in some cases may be more convenient when using certain training algorithms where *b* is derived automatically. (This applies to all subsequent functions.)



$x = \Sigma\, i_i w_i$	*Output f(x)*
$< T$	0
$\geq T$	1

fig. 3.3 thresholding function without bias input term



$x = \Sigma\, i_i w_i + b$	*Output f(x)*
< 0	0
≥ 0	1

fig. 3.4 thresholding function with bias

It might be perceived that networks composed of such cells could only be capable of producing binary outputs. However within a network of several interconnected cells

> " ... frequency of firing varies and can be viewed as a signal of greater or lesser magnitude. This corresponds to looking at discrete time steps and summing all activity ... at a particular moment in time." [Faus94, p5]

Thus the signals may also be considered as varying in level and not necessarily as being limited to fixed binary levels.

As a discontinuous function the threshold function is not viable for use with gradient-based training methods that require the use of derivatives. Thus a flattened, "S"-shaped or

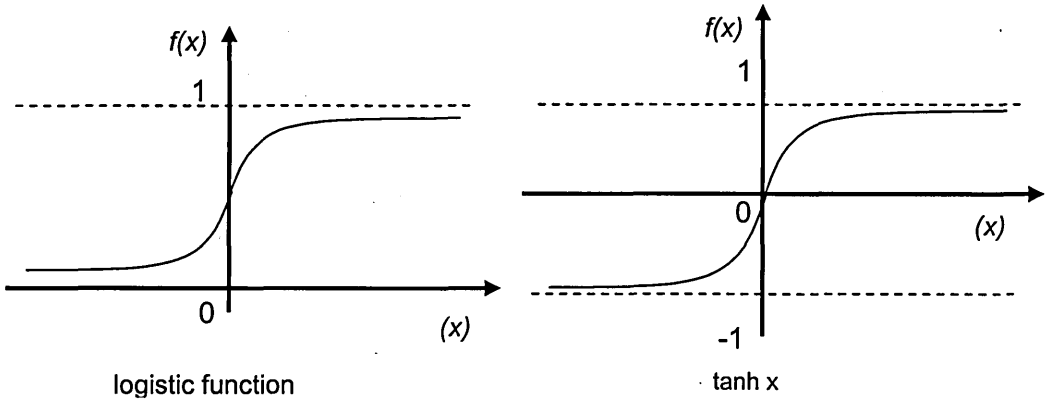
'sigmoidal', differentiable version of the function emerged, the sigmoid , or logistic, function shown below. Schalkoff [Shalk97, p78] asserts that

> "... it has a biological basis. The average frequency of biological neurons, as a function of excitation, follows a sigmoidal characteristic."

For the log sigmoid, or logistic function, $f(x) = \frac{1}{1+e^{-x}}$ —— (3.3)

where x is the sum of the weighted input signals and bias for the neuron.

Similarly for the tanh function, $f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$ —— (3.4)



logistic function tanh x

fig. 3.5 sigmoidal activation functions

The linear function is often used in the output layer of a neural network and together with the associated weights and bias(es) scales the output signal(s).

$f(x) = x$ —— (3.5)

3.1.1 Differentiation of Activation Functions

Some training methods require the use of the first derivative of the cell activation function. For the linear function,

$f'(x) = 1.$ —— (3.6)

In the case of the sigmoidal function,

$f(x) = \frac{1}{1+e^{-x}}$ $\qquad$ $f'(x) = \frac{e^{-x}}{\left(1+e^{-x}\right)^{2}}$ —— (3.3) and (3.7)

requiring the evaluation of e^{-x}, normally via a computationally-intensive series approximation. However it may be shown [Shalk97, p80] that in this particular case

$$f'(x) = f(x)(1 - f(x)) \quad \text{——} \quad (3.8)$$

with consequent savings in computational time and memory requirements.

3.2 Network architectures

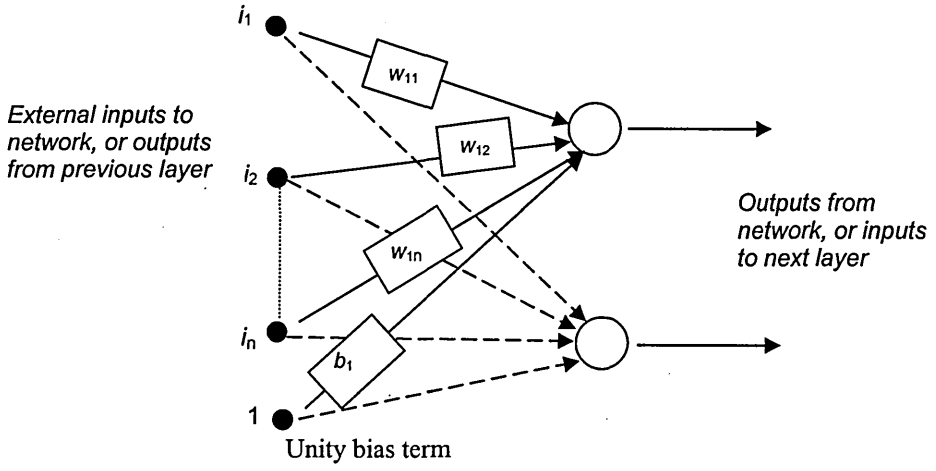


fig.3.6 example of two-cell neuron layer

Several cells may be connected in series, or 'stacked' vertically within a *layer*; these layers in turn may be connected in series with other layers. Fig. 3.6 shows such a two cell (neuron) layer with n inputs, and with the weights and bias shown for the first cell, those to the second cell being omitted for clarity. The nomenclature used is that w_{ji} refers to the weight connecting the i^{th} input to the j^{th} cell, and b_j its bias. The bias input may be considered to be an additional, constant, unity input, sometimes termed i_0 with synaptic weight w_{j0} instead of b_1. This is particularly convenient when matrix algebra is employed within training algorithms.

All cells have connections to all network inputs, or in the case of cells in a hidden layer, to all outputs in the previous layer. The option of a constant 'unity' input provides a possible bias source for the neurons.

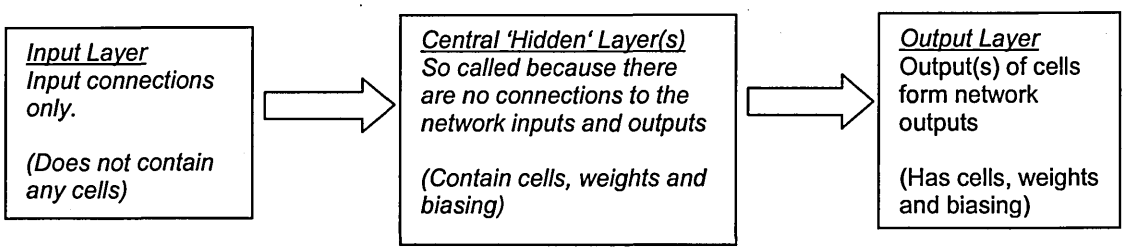


fig 3.7 schematic of feed-forward network

Fig. 3.7 shows the basic structure of a feed-forward network (FFN). (There appears to be some disagreement over the terminology in that some texts refer to this type of network as a Multi-Layer Perceptron (MLP), while others restrict the term 'perceptron' to vision applications.)

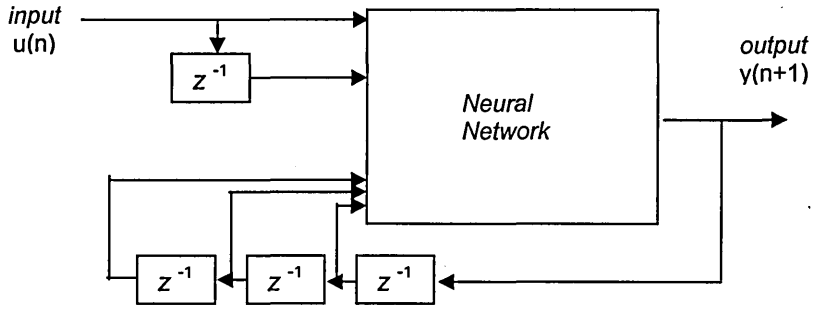


fig 3.8 external connections to network, including feedback

Figure 3.8 shows the external connections, including feedback, to a recurrent temporal network of the type used in this research. Only a single input $u(n)$ is shown together with a delayed version of itself $u(n-1)$ forming a second network input. Several such inputs may be used, with or without delays. Similarly, delayed versions of the output $y(n+1)$ form additional, feedback, inputs $y(n)$, $y(n-1)$ ... Multiple outputs may also be present.

During the course of the experimental work several variations of the above model were used, mainly one which utilised fed back versions of the output as the only inputs.

3.3 Typical Applications

Schalkoff [Schalk97, p6] characterises problems suitable for solution by neural networks as including:

> "Emulation of biological computational structures may yield superior computational paradigms for certain classes of problems. Among these are ... labelling problems, scheduling problems, search problems, and other constraint problems; the class of

pattern/object recognition problems, notably in vision and speech understanding; and the class of problems dealing with flawed, missing, contradicting, fuzzy or probabilistic data. These problems are characterised by some or all of the following : a high dimensional problem space; complex, unknown, or mathematically intractable interactions between problem variables; and a solution space that may be empty, contain a unique solution, or (most typically) contain a number of (almost equally) useful solutions ..."

Thus for example, Fausett [Faus94, p7 et seq] suggests that neural networks are employed in situations including

- pattern recognition, pattern association — particularly in machine vision applications, and signal processing generally.
- Languages — speech recognition (can be trained to recognise individual speech characteristics, hand writing analysis.
- Function approximation, particularly when information and/or data are missing or deficient., e.g. system modelling, control systems, predictors. [eg see Nar90]

Of greatest interest to this research is function approximation — the potential for a network to 'learn' the relationships between given sets of input and output data and, in particular, to operate as a predictor. Such implementations, in addition to the work describe in Section 2.0, also include prediction of tool wear [Elan95] and the control of rolling mills where data deficiency is encountered. [Rösch92]

Finally, the above solutions may be implemented as adaptive networks, where further network training occurs in real-time, between sampling intervals, so that the network responds and adapts to changes in data characteristics and system circumstances. Whilst intuitively attractive as a solution to some of the problems encountered in the experimental work the following factors precluded the inclusion of adaptive solutions:

- There exists the potential for the network to 'forget' the longterm historical data characteristics.
- There is the possibility of a pre-trained network 'adapting' from a generalised solution to a 'memorised' one. (Section 3.4.2)
- Training requires considerable computing resources; the preferred solution was a minimalist one to be implemented on relatively unsophisticated hardware.

3.3.1 Function Approximation [e.g. Hayk99. P209; Shalk97]

This is a development of the Stone Weierstraß theorem, which Haykin [Hayk99. P209] interprets as

$$F(x_1,.....,x_{m_0}) = \sum_{j=1}^{m_1} \alpha_j \psi\left(\sum_{i=1}^{m_0} w_{ji} + b_j\right) \qquad \text{——————} \quad (3.10)$$

where $\varphi(.)$ is a "nonconstant, bounded, and monotone-increasing (sic) continuous function."

as an approximate realisation of the function f(.), i.e.

$$\left|F(x_1,.....,x_{m_0}) - f(x_1,.....,x_{m_0})\right| < \varepsilon \qquad \text{——————} \quad (3.11)$$

for all $(x_1,.....,x_{m_0})$ that lie in the input space and where ε is the permitted error,

(The logistic and tanh activation functions both fulfill these criteria.)

Equation (3.10) maybe interpreted as describing a network having m_0 inputs, a single hidden layer of m_1 non-linear neurons with connecting weights w and bias values b and with linear outputs of synaptic weighting α_1 .... α_{m1}.

Effectively, a single hidden layer is sufficient to enable a network to compute an approximation to any given function.

Significantly, Haykin concludes

> ".... However, the theorem does not say that a single layer is optimum in the sense of learning time, ease of implementation, or (more importantly) generalization."

3.4 Practical considerations in network implementation

3.4.1 Network design

Fundamental to network design is the question of the number of layers in a network and the number of cells per layer. All networks must have an input and an output layer. The function of hidden layers is to provide feature identification; Haykin [Hayk99, p199] states that

> "... As learning progresses, the hidden neurons begin to gradually 'discover' the salient features that characterise the training data. They do so by performing a nonlinear transformation on the data into a new space called the *hidden space,* or *feature space* ..."

Schalkoff [Shalk97, p166] elaborates, referring to hidden layers as internal layers:

"1. The internal layers *remap* the inputs and results of other (previous) internal layers to achieve a more seperable representation of the data. Suitable external preprocessing of the inputs may yield the same effect.

2. The internal layers may allow the attachment of semantics to certain combinations of layer inputs and thus serve as matched filters. Often it is illustrative to examine the structure of internal layer remapping that evolves from training. For example, in investigating the neural network application to edge classification, it was observed that hidden units behaved like 'feature detectors', each implementing a portion of a matched filter." ... (memorisation) ..."may be an extreme case of this behaviour."

The number of cells in the output layer is governed by the number of network outputs required by the application, as is the number of inputs. However deciding the number of cells in a hidden layer is a somewhat iterative, heuristic process. Schalkoff [Shalk97, p166] commences his discussion of the topic with the observation that

"The choice of the number of hidden units in a feedforward structure design often involves considerable engineering judgement. Often, trade-offs between training time and mapping accuracy lead to iterative adjustment of the network using simulation. For a given problem, the design of an appropriately sized hidden layer is often nonobvious. Intuition suggests that 'the more the better' could be used as a guide to sizing of the hidden layer, since the number of hidden units may be counterproductive. For one thing, the network training time is influenced by the size of the hidden layer ... Increasing the number of hidden units greatly ... increases training time substantially, with little gain in overall mapping capability of accuracy. Furthermore, an excessively large number of hidden units allows an undesirable" ... (memorising) ..." effect which results in units memorising certain aspects of certain units rather than providing a distributed computation."

Haykin [Hayk99, p199] suggests two approaches : *network growing* and *network pruning* appearing to favour the latter.

Network growing : the design process commences with a minimal network and then adding further cells and layers only when it is established that the required degree of accuracy can not be met without enlarging the network.

Network pruning : the design process commences with a large network

"...with an adequate performance for the problem at hand, and then prune it by weakening or eliminating certain synaptic weights in a selective and orderly fashion..."

until there is a significant reduction in accuracy.

The latter approach requires a large amount of memory and a high speed processor to offset the increase in training time. Given the restricted computing power and capability available to the project it was decided to follow a network growing approach. Further both Reinschmidt and Boccaletti [Section 2.0] describe network growing approaches in their accounts of practical network development.

Interestingly, Haykin appears to infer a minimum of two hidden layers when discussing the practical considerations of using a network as function approximator, and states that local features (of the data set) are extracted in the first layer whilst global features are extracted in the second [Hayk99, p199]. (Inferring perhaps, that the global features are necessary for generalisation.)

3.4.2 Generalisation and memorisation

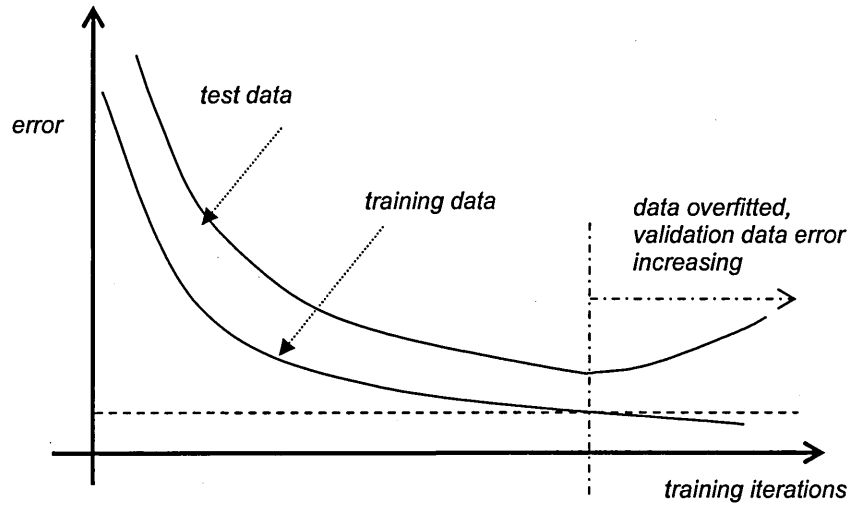


fig. 3.9 validation (test) set error increasing after 'over-fitting'

A significant issue in design and training is that of generalisation verses memorisation. The available data is divided into two sets, a training set to actually train the network by establishing the synaptic weights of the network, and a separate test, or validation, set to verify the network's performance and accuracy.

Generalisation refers to the ability of a network to perform accurately with input data other than that presented to it during training. If training is allowed to continue until the output error of the network when presented with the training data reaches a minimum, memorisation may occur. That is, the network eventually memorises the training set input-output relationships and behaves as a look-up table, losing its 'knowledge' of the data characteristics in the process. When (previously) unseen data is presented to the network, e.g. for validation purposes, it may not produce the required outputs.

Both Schalkoff and Haykin advocate the frequent interruption of the training process and validation of the results so far obtained, in order to avoid 'overtraining'. Whilst there might be scope for further error reduction with the training set, the network's output error with the validation data begins to increase. (fig. 3.9) In fact Schalkoff [Shalk97, p195] raises the possibility that an extreme case might be that the network performs poorly with the training data but performs well on validation. In practical terms, Schalkoff [Shalk97, p196] suggests that available data be split 20% for training data and the remainder used for validation.

3.4.3 Preprocessing of Data

Rao [Rao95, pp379-381] in his case study of a financial forecasting model raises several issues of relevance to this research, grouping his comments as follows:

- Highlighting of (input data) features
- Transformation
- Scaling and bias (offset)

He suggests consideration of presenting rates of change (first and second derivatives) of the data as additional inputs to the network. (The experience of this writer during the work reported in this thesis was that negative weightings appear during network training and that these, combined with the remaining positive ones, combine at the summing junction to give difference (i.e. derivative) information to the network, and hence such inputs are largely superfluous. [Section 5.3.7])

Further it might also be beneficial to emphasise, or highlight, certain important influences by using binary inputs to represent their presence or otherwise. (In his example on financial systems Rao cites Central Bank intervention.) Furthermore it might be beneficial to "de-emphasise" unwanted noise in the inputs, e.g. data spikes, by smoothing it with a moving average filter. However he cautions against "introducing excessive lag in the resulting data."

For time series data Rao advocates consideration of the use of a Fourier Transform to expose cyclic events, together with other signal processing techniques such as digital filtering.

Recalling the 0 to 1 output range of the sigmoidal activation function, he recommends that data should be chosen that does not cause saturation, or "overwhelm" the neurons. He suggests normalising both inputs and corresponding outputs. Furthermore, given that the GDR [Appendix A.2] is less effective for near-zero training values the addition of a constant bias might make training more effective. (0.5 is suggested for a sigmoidal function to bring the data near to the centre of the output function. However, in the Matlab neural network toolbox, and other systems, bias values are computed during training rather than implemented as fixed constants.)

Rao does not comment on the tanh sigmoidal function which is antisymmetric about zero, and which finds favour with Haykin:

> "A multilayer perceptron trained with the back-propagation algorithm may, in general, learn faster (in terms of the number of training iterations required) when the ... " (activation function) ... " is antisymmetric than when it is nonsymmetric ... A popular example of an antisymmetric activation function is ... in the form of a *hyperbolic tangent* ... "

In contrast to Rao, Haykin later adds (with respect to the hyperbolic function)

> "Each input variable should be *preprocessed* so that its mean value, averaged over the entire dataset, is close to zero, or else it is small compared to its standard deviation."
>
> [Hayk99, p179-181]

3.4.4 Training

Standard back-propagation [Appendix A.2] was adopted for the early stages of this research due to the extremely limited computing resources then available. This proved to be somewhat slow and inefficient, requiring several hours training for a single network. However in the latter stages when much improved facilities did become available the Levenberg-Marquardt variation was adopted with much improved timing and efficiency.

The Levenberg-Marquardt algorithm involves numerous matrix inversions and as such is computationally demanding; however it is described as the fastest training method available [Hagan95: chapter 12].

4.0 Experimental strategy and procedures

4.1 Background and analysis

4.1.1 Software used

MATLAB was used throughout the project, in particular the MATLAB neural network toolbox. Other systems were available, principally SNNS (Stuttgart Neural Network Simulator). However this required computer operating systems that were not readily available to the project - e.g. Linux or Unix - or the installation of X-Windows to enable the software to run on the Microsoft Windows platform. [SNNS98]

4.1.2 Accuracy considerations and benchmarking

Corus had considered the use of a second-order least squares-based predictor. Hence such a predictor was implemented in MATLAB and its performance used as benchmarking for neural network development.

Accuracy measurements were made on a per unit basis:

$$e = \left| \frac{predicted - actual}{actual} \right| \qquad \text{—— (4.1)}$$

Further, for a given experimental run, the mean and maximum error values were recorded, together with the standard deviation as an indication of the dispersion of the results.

4.1.2.1 Effects of pre-processing on accuracy

Pre-processing may be employed in order to bring the data ranges closer to those of the non-linear activation function sigmoids, i.e. <= 1.0, thus reducing the magnitude and number of weight changes during backpropagation (Appendix A2) , and enhancing the efficiency and speed of training.

A method of pre-processing was investigated, involving division of the current data value by a running mean of the last *m* values, in accordance with the following algorithm:

$$x'[k] = \frac{x[k]}{\bar{x}[k]} \qquad \text{—— (4.2)}$$

where $x'[k]$ is the pre-processed value

$x[k]$ is the actual value

$\bar{x}[k]$ is the current value of the running average

and $\bar{x}[k] = \frac{1}{m} \sum_{n=k-m+1}^{k} x[n]$ for the previous m values —— (4.3.)

However at 'post-processing' after predicting $x'[k+1]$, the value of $\bar{x}[k+1]$ will not be known, so the data will have to be re-constituted using $\bar{x}[k]$ as shown below:

$$x[k+1] = x'[k+1] * \bar{x}[k] \quad \text{—— (4.4)}$$

with consequent inaccuracies (particularly where data is subject to rapid fluctuations relative to the sampling interval) in addition to those resulting from network training, and thus placing limitations on the maximum possible accuracy attainable. Further, division would have a smoothing effect on the data introducing a loss of dynamics which might reduce the effectiveness of the training process.

It should be noted that an advantage of pre-processing by division by a running mean is that a dimensionless dataset is produced which is independent of the measurement units, and further, where the absolute ranges are not known, reduces the range of the datasets.

Simple scaling through division by 10 or 100 was also used. This requires little in the way of post-processing and preserves the dynamics of the data. (Significantly higher accuracies were obtained by scaling than by preprocessing through division by a running mean.)

4.1.3 Neural network-specific issues

With reference to Section 3.4.1, few texts give adequate indications of possible sizes and structure (i.e. the number of network inputs, the number of layers, the number of cells per layer, and the type of activation functions) of networks for given applications, particularly for a predictor network involving MLPs, other than experimental iteration. However, Reinschmidt [Rein95a] suggests a method for determining the approximate number of inputs, which was adopted as a basis for the work described in this thesis (described in section 2.3). However, his discussion gives little indication of the type of non-linear activation function and hence, both types of sigmoidal function were investigated during the relevant stages.

4.2 Strategy

It was decided that Reinschmidt's neural network-based work as discussed earlier in Section 2.3 might form a suitable basis for investigation and that development should proceed from there. However, it should be noted that his paper

- reported the development of a load forecasting system rather than fuel parameters and that the system under investigation was a conventional single-fuel power generation plant rather than mixed fuel system of fluctuating content.
- concerned hourly predictions and that this project required predictions on a minute-by-minute basis, and possibly more than one sampling interval ahead.

Reinschmidt describes three approaches – a linear neural network, a genetic algorithms method, and a non-linear neural network. All of which are adaptive. Further, the limiting time for any computations is a one-hour sampling interval, which allows scope for some lengthy and sophisticated computations.

In this work, in the case of predicting one interval ahead, there would be less than one minute due to the time required for mass spectrometer computations and valve train operation, and the solution implemented on equipment with only limited computational resources, such as low-specification PCs or PLCs redundant from other projects It was therefore decided to proceed with a non-adaptive approach and a purely neural network solution, using both linear and non-linear techniques, rather than incorporating genetic algorithms-based work which is extremely demanding in terms of resources. (Section 1.1)

4.2.1 Proposed network structures

It was proposed to develop linear networks based on the method outlined in Section 2.3 (Equation (2.1) reproduced as equation (4.5) below).

$$z(t) = a_1 z(t-1) + a_2 z(t-2) + \ldots + a_n z(t-n) + u(t) \quad \text{—— (4.5)}$$

where z is the predicted (tracked) parameter

a_n etc, are coefficients

$u(t)$ is a random disturbance

Equation (4.5) is analogous to a single layer, single cell network with n inputs and with a linear activation function, and whose output is given by

$$y(p) = w_1 y(p-1) + w_2 y(p-2) + \ldots + w_n y(p-n) + b \quad \text{—— (4.6)}$$

where w_n, etc are input weightings

b is a bias added to the sum of the input signals, and whose value is determined during training.

n may be determined during network development by allowing the number of inputs to 'grow' from a predetermined lower limit to an upper one, and observing the number of input weightings with low values – i.e. those with potentially extraneous inputs. This would provide an indication of the optimum number of network inputs. Where relevant, this was used as a basis for non-linear network development. It was felt that a lower limit of 3 would be the minimum required to facilitate the possible inclusion of gradient (rate of change) information. (Weightings of different signs would indicate that subtraction was taking place and first-order difference information had been absorbed into the network.)

4.3 Procedures

Data from the mass spectrometers monitoring gas content was supplied in the form of spreadsheets which were then converted to a text format readable by MATLAB. Script files were developed which automated loading of data and preparation of datasets for training, validation and verification during training, and introducing any required pre-processing.

The main datasets were divided into smaller subsets with the subset possessing the widest dynamics being used for training and the subset with the second most extreme excursions being used for validation. The remaining data points were then used for verification purposes. (The justification for this being that the network would be trained with the 'worst case' available, in the absence of any other information.)

Further script files were developed which automated the training process, iterating through a chosen range of inputs (usually 3 to 20) for a particular type of network and, in the case of non-linear networks, further iterating through a prescribed number of cells in a hidden layer for a particular number of inputs. Training utilised standard back-propagation [Appendix A2] initially until the acquisition of more sophisticated hardware permitted the introduction of the superior (in terms of training time) Levenberg Marquardt adaptation. [Section 3.4.4]

The MATLAB neural network toolbox was used to produce feedforward networks which were then simulated against verification data, and the maximum and mean absolute errors were recorded, together with the standard deviation. Furthermore, when it was envisaged that a linear network would be used to indicate the optimum number of inputs for non-linear solutions the number of input weightings less that 0.01 in magnitude was recorded as an indicator of potentially extraneous inputs.

Initially there was some uncertainty in respect of the possible available hardware for implementation, and it was believed that a low-specification PC might be available. Hence some non-linear solutions were investigated, albeit generally with a limited number of hidden cells in keeping with a minimalist solution to reduce computational overheads. In these cases both types of sigmoidal activation function – logistic and tanh (MATLAB refers to them as 'logsig' and 'tansig') – were employed.

For the CPS work only one parameter – calorific value (CV) – was to be predicted. Later work introduced the additional parameters of specific gravity (s.g.) and air/fuel ratio (a/f) were introduced as were Wobbe numbers (or indices) for both phases of the HSM work. Where a parameter was not directly available, data was synthesised using appropriate formulae. Data sampled at 1-minute intervals was also synthesised from either that sampled at 23-second or 30-second intervals directly at the mass spectrometer outputs for comparison purpose, i.e. to induce the effects of aliasing inherent in the main archiving system.

Where parameter ranges were greater than 1.0 – the conventional upper limit of non-linear neuron activation functions – an additional linear output cell was added to non-linear networks for output scaling purposes. However, the need for the further output cell and layer could be avoided by introducing pre-processing of the data by scaling through division by a constant – e.g. 100.

A further method of pre-processing was investigated, that of dividing the current data value by a running mean of the last m values. This had the advantage of bringing the values in different datasets into a common ranges around unity, thus facilitating training of generalised networks when the overall data ranges likely to be encountered are unknown. It was established that $m = 2$ produced the most acceptable results, and in this text the expression '$m = 2$' refers to networks trained with this method. In that case, the effects of post-processing are also tabulated with relevant absolute error parameters. However, it was later established that for rapidly changing data – e.g. in the HSM – the results were greatly inferior to the other pre-processing methods and it was abandoned. In fact there proved to be little difference if any, between developing linear networks without pre-processing and those pre-processed through division by a constant so that ultimately, development of linear networks proceeded without any pre-processing.

The HSM data was subject to extreme and abrupt fluctuations in parameter levels when switching from one mixing station to another. [Chapter 6] CO and H_2 levels acted as potential

indicators as to which station was the dominant supplier. Linear networks which contained additional inputs of CO and H_2 were also investigated.

4.4 Summary of procedures/experimental steps

The above experimental procedures may be summarised as follows

1. Examination of the supplied spreadsheets to obtain data ranges, mean and median values, and their standard deviation.

2. Selection of sets for network training, validation and verification purposes. Given the relatively short time spans of the supplied data with respect to the plants' all-year operations, the data set with the maximum range and dynamics were chosen for training to give maximum training efficiency; those with the least were selected for verification.

3. Network development proceeded using non-preprocessed data followed by data pre-processed by scaling, and then by the '*m*=*n*' running mean method of processing.

4. Maximum and mean absolute error together with the standard deviation of the absolute error were noted. Tables of typical results are shown in each section; for the non-linear work the 10 network solutions with the lowest maximum absolute error are listed.

5. Where a non-linear network investigation was conducted, network architectures were based on the corresponding linear network results both in terms of potentially extraneous inputs and network accuracy.

6. Potential network solutions were assessed both with respect to their overall accuracy and in terms of computational requirements. There were compromises between reduced accuracy and complexity.

4.4.1 Notation used when reporting neural network results

The following abbreviations and conventions were used to describe network structures:

K = linear activation function

L = log sigmoid, or 'logsig', activation function

T = tanh sigmoid, 'tansig', activation function

E.g. 6+K describes a simple linear network with 6 inputs. 7+2L+K describes a 7-input non-linear network with 2 'logsig' cells in a single hidden layer, with a linear output (scaling) cell.

Where pre-processing was investigated the abbreviations below indicate the method:

'm = n' Scaling through division by m, where m is the mean of the past n values

'$\div k$' Scaling through division by a constant.

E.g '$m = 2$', or '$\div 100$'

* Potentially optimum solutions in terms of accuracy and/or minimalist structure are indicated in the tables by an asterisk.

5.0 Central Power Station (CPS)

The Central Power Station (CPS) operates as a cogeneration plant supplying steam and electricity for on-site processes, and hot water for office heating, but electricity generation is sacrificed when demand for 'process steam' is high.

The plant contains a diversity of equipment, much of it acquired as surplus from other BS sites around the UK. The availability of that equipment has dictated its development to a significant extent, and as a result has its own particular requirements and problems.

There are several boiler/furnace combinations which supply steam via common mains to a number of turbo-generator sets. Unlike the HSM (Section 6.0), the different constituent fuels appear to have separate burners and mass spectrometers. The boiler designs are a compromise from the ideal because of the need to burn any of, or a combination of, the following fuels (see also Appendix A.1)

- Coke Oven Gas (COG - mainly Hydrogen and Methane)
- Blast Furnace Gas (BFG - mainly Carbon Monoxide, Nitrogen, and Carbon Dioxide)
- Synthetic Coke Oven Gas (SCOG - which is produced by mixing BFG with bought-in Natural Gas to produce a gas similar to COG)
- Heavy Fuel Oil

Both BFG and COG are on-site by-products and are negligible in cost compared to the price of the 'imported' fuels. Thus maximising their use in boiler operation is of particular importance.

COG is heavily contaminated and the resulting fouling of flues also affects the operation over time. Additionally all boilers are equipped with COG pilot burners so some COG is always being burnt.

Some method of predicting calorific values as an aid to optimising the control systems and overcoming some significant transport lags caused by the measuring equipment being situated some distance from the burners and the associated valve trains being somewhat slow in operation was investigated. (It was understood that the valve trains took some 20+ seconds to operate.)

5.1 Analysis of data

The following data files were extracted from the supplied spreadsheets containing information from the BS logging system. Values of CV are given in MJ/kg for range, mean and median.

usage	*from*	*to*	*points*	*range*	*mean*	*median*	*s/dev*
Training	22:02:00 10/May/99	06:03:00 11/May/99	482	18.03 – 22.09	20.65	20.81	0.94
validation	22:00:00 17/May/99	06:01:00 18/May/99	482 (369*)	17.42 - 19.87*	18.59	18.55	0.56
verification	23:00:30 10/Jul/99	07:00:00 11/Jul/99	480	17.07 – 20.29	19.28	19.49	0.81

* = Has zero values between 74 and 114 mins. Some form of equipment failure is assumed. Range for 114 mins onwards quoted.

Table 5.1 Properties of supplied data with intended usage

Examination of the above files showed that several files covering a given period have an irregular sampling intervals (30 secs instead of 1min. at midnight) and therefore theoretically should not be used to make one continuous data file. However, information received confirmed that aliasing was already present in the data (the mass spectrometers operated at 25 – 35-second intervals) and therefore such a combined file might be used with little or no effect on training and accuracy as a result of the irregular interval at the join.

Figures 5.1 to 5.3 show variations in values of CV for the three usages shown in table 5.1.

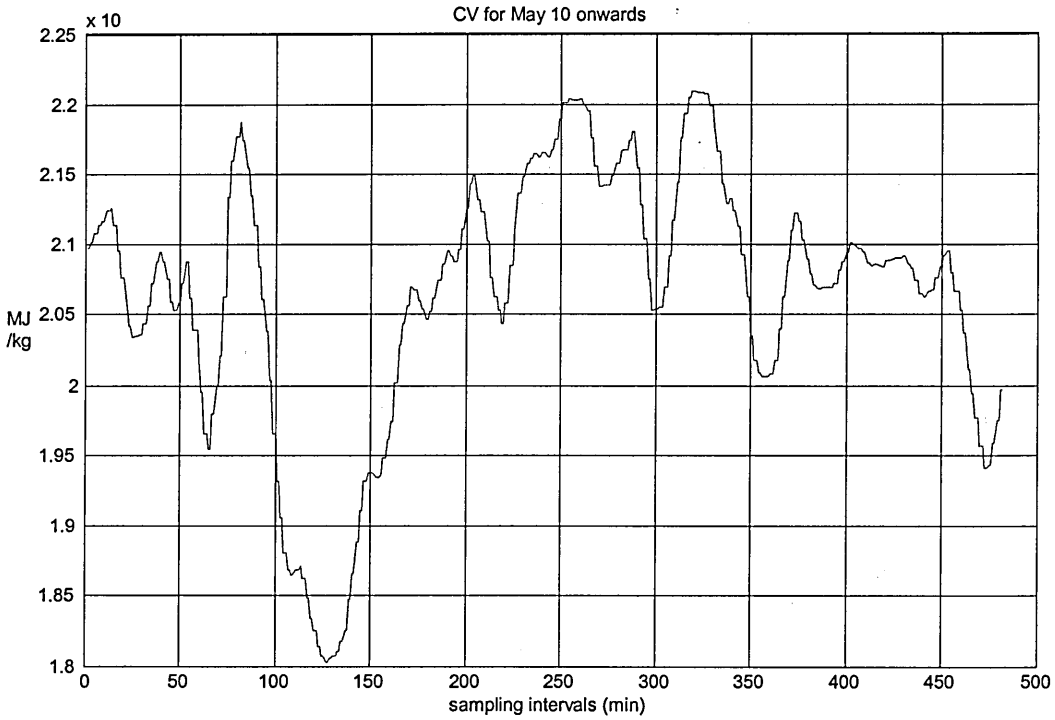


fig. 5.1 CV May 10th onwards

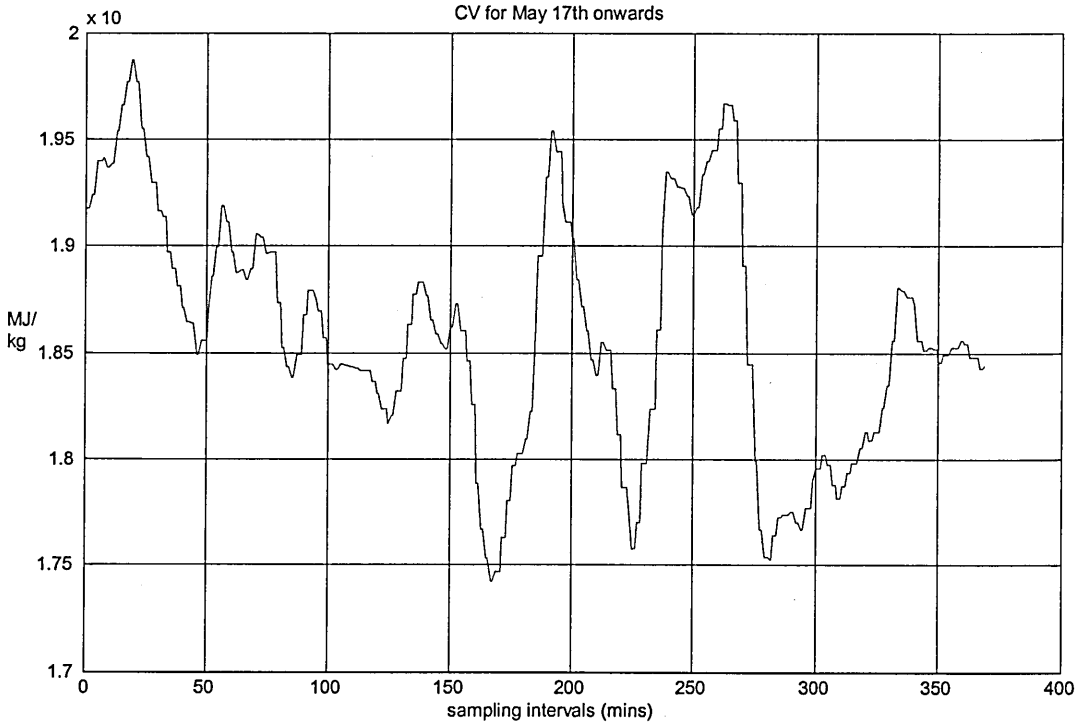


fig. 5.2 CV May 17th onwards

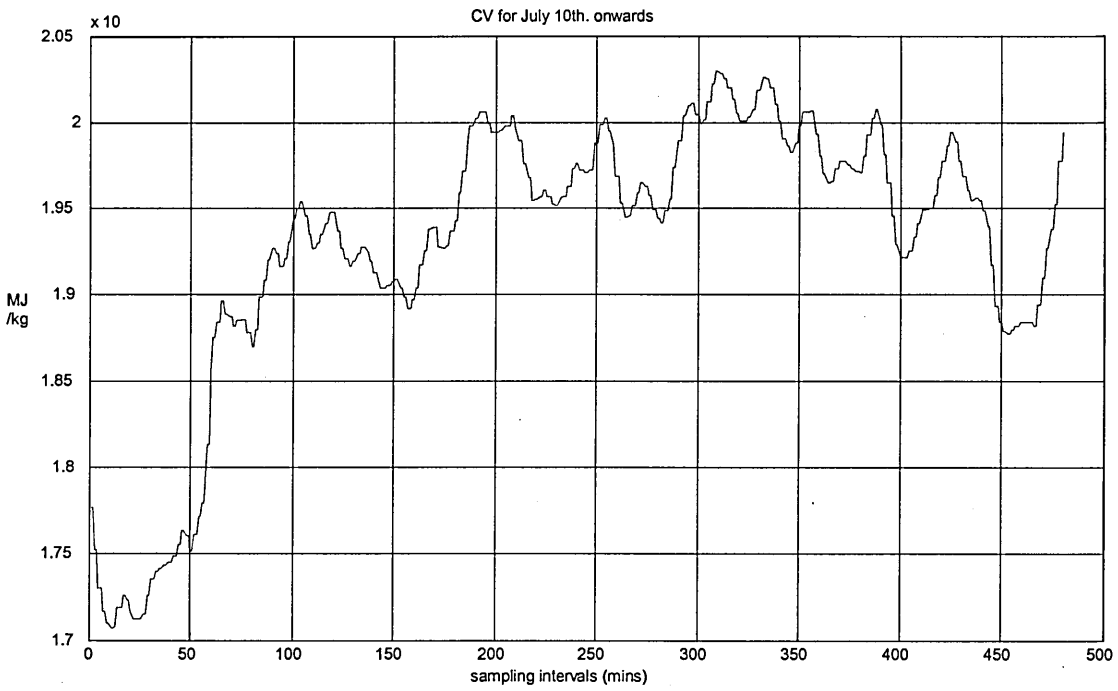


fig. 5.3 CV July 10th onwards

There are potentially significant differences in ranges which might be overcome by pre-processing.

percentage constituents		*range*	*mean*	*Appendix A.1*
Carbon monoxide	CO	0 – 10.08	5.69	6
Carbon dioxide	CO_2	1.35 – 3.05	2.4	2.41
Hydrogen	H_2	50.21 – 65.56	58.09	51.47
Nitrogen	N_2	0 – 12.65	3.81	7.17
Methane	CH_4	22.07 – 31.65	26.63	29.94
Ethylene	C_2H_4	0.59 – 1.13	0.95	1.67
Ethane	C_2H_6	1.33 – 1.95	1.71	0.82
Calorific value (MJ/Nm^3)		17.07 – 22.09	19.03	19
specific gravity		0.35 – 0.35	0.35	0.38
Air/fuel ratio (true)		Not supplied	not supplied	—

Table 5.2 Comparison of supplied data with typical values in Appendix A.1

Table 5.2 describes the main gas characteristics found in the above files, and how they compare with the typical COG values given in Appendix A1. Whilst some parameters are similar, CPS COG is higher in hydrogen, lower in methane, and significantly lower in nitrogen. Further the values for ethylene and ethane vary by approximately 50%. Mean calorific value and specific gravity are at typical levels.

5.1.1 Effects of pre-processing of data by the mean of the previous m *values*

Pre-processing of the data through division by a running mean of the last m values alters the dynamics of the data and introduces errors when reconstituting the data during post-processing. For an ideal network with 100% accuracy in prediction, it is possible to calculate the errors introduced in post-processing for various values of m by pre-processing the datasets and then reconstructing them according to equations (4.2) to (4.4) assuming the predictions to be 100% accurate.

dataset	*m = 2*	*5*	*10*	*20*
training	1.68 (0.14, 0.26)	2.44 (0.29, 0.43)	4.63 (0.89, 1.17)	6.53 (1.17, 2.27)
Validation	1.24 (0.14, 0.27)	1.92 (0.30, 0.44)	4.21 (0.84, 1.15)	6.89 (1.50, 2.10)
Verification	1.17 (0.09, 0.17)	1.56 (0.19, 0.28)	3.64 (0.52. 0.70)	5.17 (0.89, 1.19)

Table 5.3 Error levels for increasing values of m (see text)

Table 5.3 shows the maximum percentage absolute error (with mean and standard deviation in parenthesis) for increasing values of m. In all cases as m is increased, there are corresponding increases in all error parameters; increasing m beyond 2 was seen as counter productive.

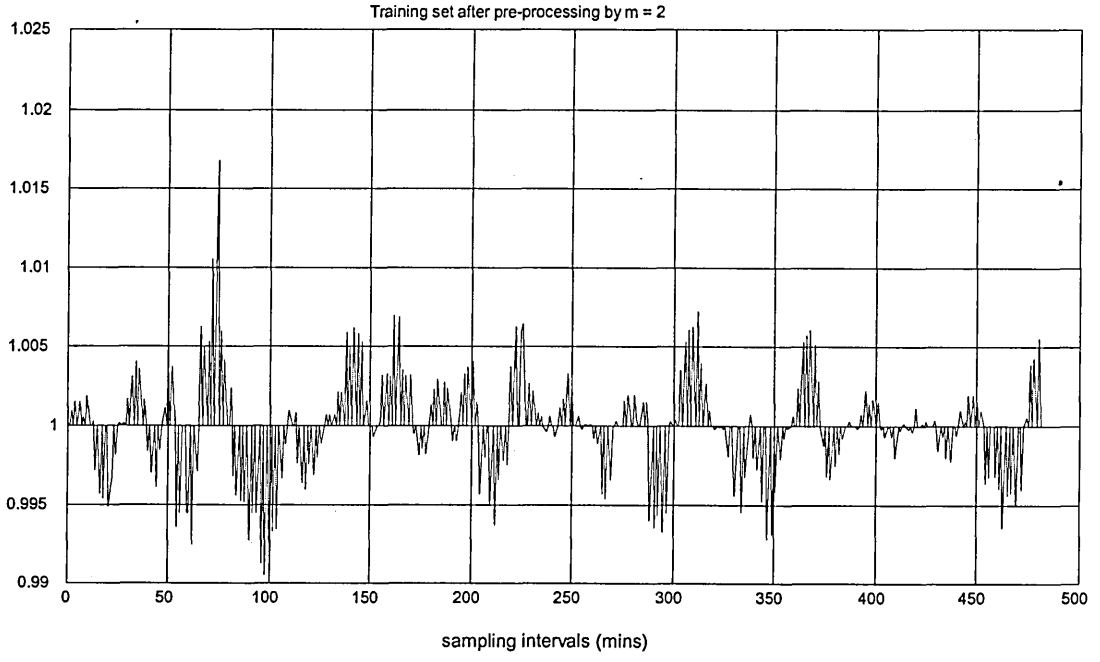


fig. 5.4 Effect of preprocessing by m = 2 on training set

Figure 5.4 shows the altered dynamics of the training set data for $m = 2$, with corresponding plots of the validation and verification sets in fig. 5.5 and 5.6.

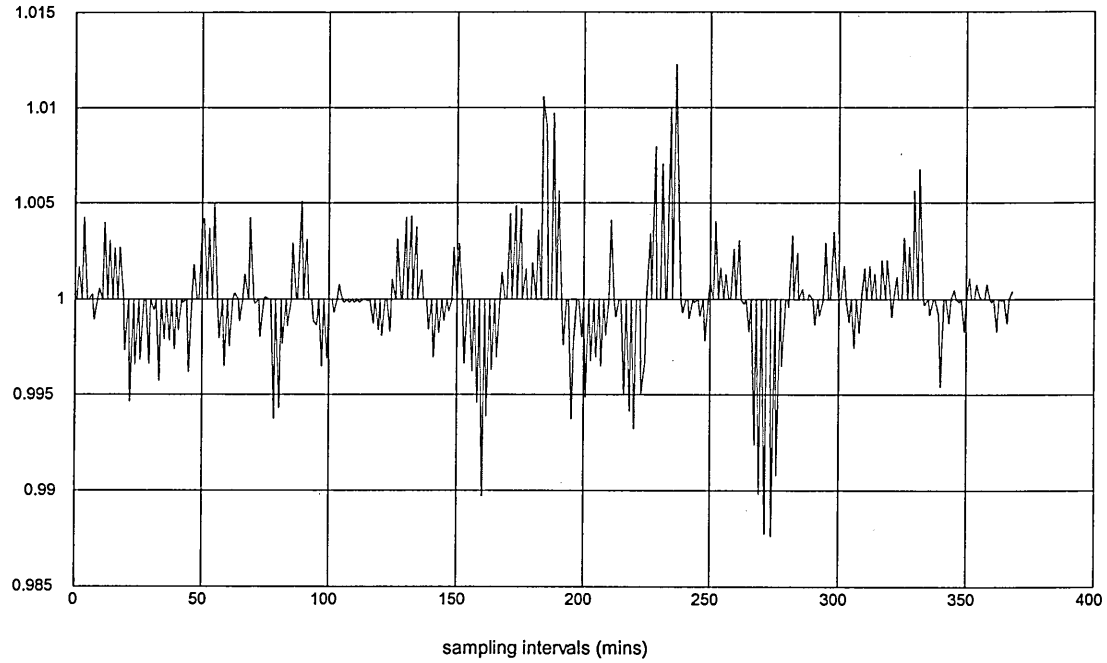


fig. 5.5 Validation set after preprocessing by $m = 2$

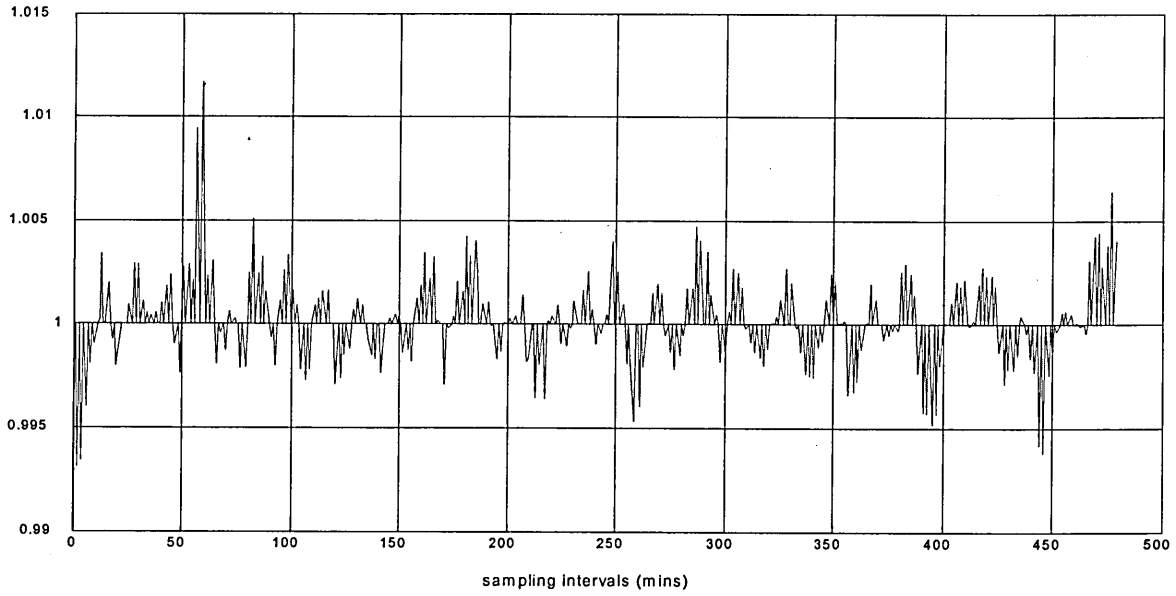


fig. 5.6 Verification set after preprocessing by $m = 2$

Figure 5.7 shows the reconstructed (post-processed) training set data for $m = 2$ which compares closely with the original plot in fig. 5.1.

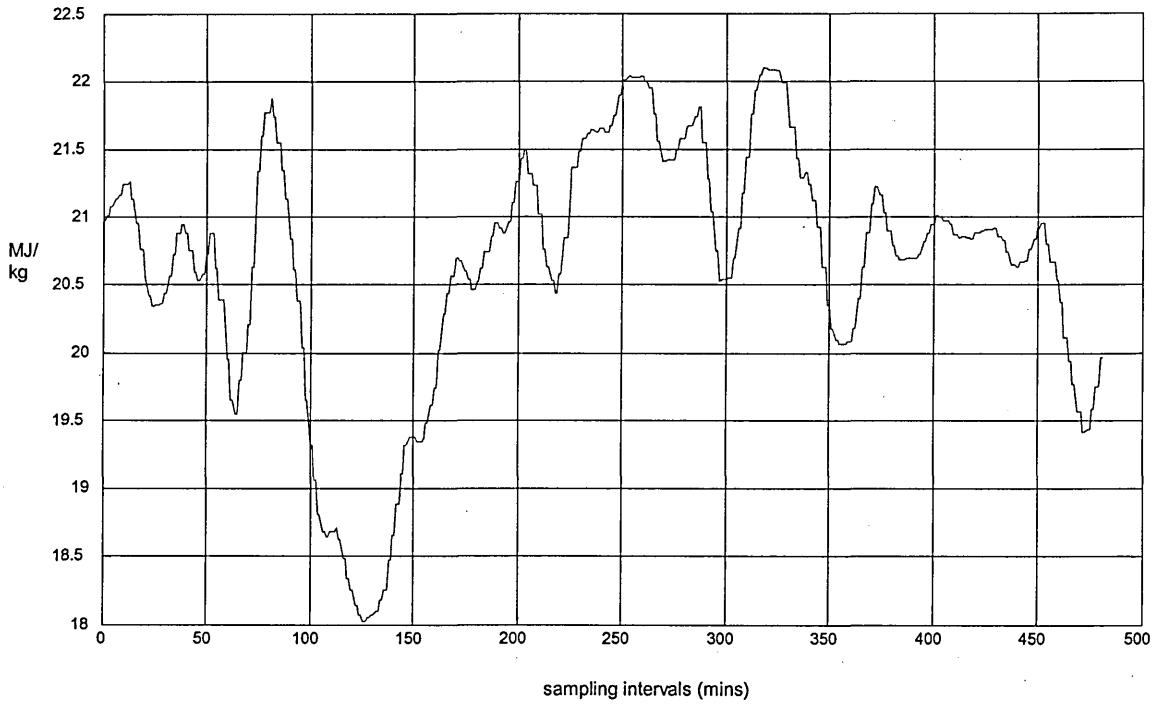


fig. 5.8 Training set (reconstituted data – see text)

The modified ranges for the data sets after pre-processing by $m = 2$ are shown in table 5.4:

dataset	*range*	*mean*	*median*	*std. deviation*
training	0.99 – 1.02	1.0	1.0	0.003
validation	0.99 – 1.01	1.0	1.0	0.003
verification	0.99 – 1.02	1.0	1.0	0.002

Table 5.4 Modified dataset ranges after preprocessing by $m = 2$

All sets now have similar ranges of values, including mean and standard deviation, and hence similar characteristics.

5.2 Overall results and discussion

Tables of typical results are given in Appendix A3; the tables below summarise those results. In addition to the most accurate results obtained, tables 5.5 and 5.6 also show potentially acceptable results which are less computationally demanding. The columns list type of pre-processing, network structure, activation function and percentage absolute error – mean, maximum, together with the standard deviation. For the '$m = 2$' systems, the post processing error is quoted.

p.p / method	*struct*	*funct*	*mean*	*max*	*s.d.*
lsq*	—	—	0.29	1.87	0.25
None	13+K	linear	0.18	1.52	0.20
÷100	11+K		0.18	1.49	0.20
m = 2	7+K		0.23	2.11	0.22
÷100	12+Kog	log	0.24	1.50	0.27
	13+tanh	tanh	0.18	1.52	0.20
None	11+4+K	log	0.18	1.32	0.20
	11+2+K		0.19	1.51	0.21
	12+3+K	tanh	0.23	1.30	0.23
	12+2+K		0.19	1.48	0.21
÷100	12+6+K	log	0.23	1.33	0.23
	12+2+K		0.19	1.71	0.21
	13+2+K	tanh	0.20	1.57	0.20
m = 2**	13+6+K	log	0.26	2.07	0.26
	13+4+K		0.25	2.19	0.26
	13+8+K	tanh	0.25	1.99	0.25
	13+2+K		0.28	2.25	0.28

*least squares benchmarking result **after post-processing

Table 5.5 Possible network solutions (predicting one sampling interval ahead)

The '$m = 2$' networks required significantly less training time but produced inherently lower accuracy due to post-processing error. Further, there was a tendency to produce maximum error levels in excess of those resulting from the benchmark least squares-based predictor. Networks with tanh sigmoid activation functions were capable of high accuracy but software instability occurred regularly during training as a result of "singular matrix" errors. Pre-processing through division by 100 also resulted in lower maximum error but at the cost of increased mean absolute error and standard deviation.

p.p / method	*struct*	*funct*	*mean*	*max*	*s.d.*
lsq*	—	—	0.29	1.87	0.25
None	13+K	linear	0.28	2.06	0.25
÷100	3+K		0.29	1.97	0.26
	12+K		0.28	2.04	0.25
M = 2	5+K		0.37	3.22	0.33
None	11+13+L	log	0.31	1.72	0.26
	12+4+L		0.31	2.03	0.26
	12+6+L	tanh	0.32	1.72	0.27
	12+3+L		0.31	1.88	0.27
÷100	12+7+L	log	0.40	1.95	0.31
	12+2+L		0.31	2.23	0.27
	11+4+L	tanh	0.35	1.72	0.31
	12+2+L		0.37	2.57	0.34
m =2**	12+7+K	log	0.40	3.11	0.36
	12+3+K		0.37	3.16	0.34
	11+7+K	tanh	0.39	3.06	0.35
	12+3+K		0.40	3.22	0.37

*least squares benchmarking result **after post-processing

Table 5.6 Possible network solutions (predicting two sampling intervals ahead)

There was, as anticipated, a general increase in error levels when predicting values for two sampling intervals ahead.

Effective results were obtained from simple linear networks with no pre-processing. None of the linear network solutions exhibited a large number of potentially extraneous inputs; a maximum of 1 per network being found to be below the arbitrary threshold of 0.01.

In conclusion it must be stressed that the above results were obtained from relatively short datasets (ca. 8hrs) which may not be typical of operations at this plant. (In fact, all the files cover night time operations; there is nothing to indicate that daytime processes are identical.) A discussion of the CPS results in comparison with those of Dawes Lane is given at the end of Chapter 7.

6.0 Heavy Section Mill

The fuel supply for the Heavy Section Mill (HSM) consists of a gas main common to a number of mills. (fig. 6.1) This main is supplied with different gas mixes via two mixing stations with a wide range of variations in gas content, and supplied via secondary supply mains to the individual mills. It is even possible for a situation to exist where a 'dead band' occurs between the two mixing stations where the gases mix, but there is zero flow. Furthermore, it is also possible at certain times that only one of the stations may be supplying all of the mills.

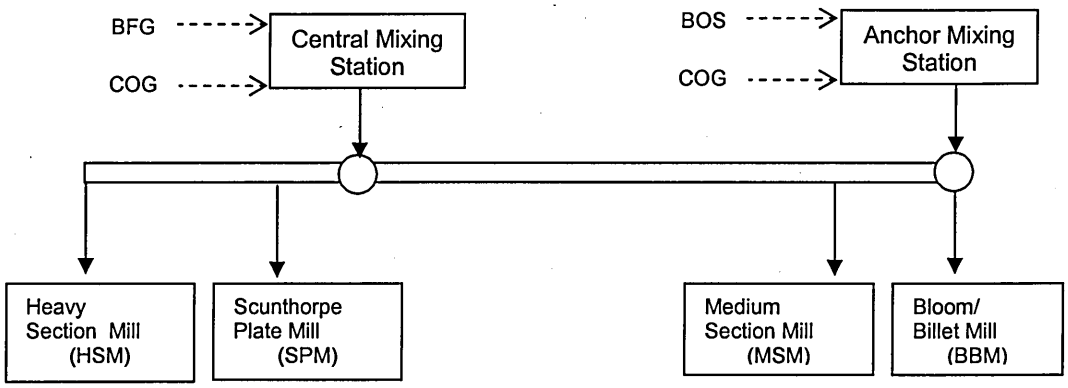


fig. 6.1 Schematic of fuel supply to HSM

The two mixing stations supply Mixed Enhanced Gas (MEG) which has different constituents depending which station supplies it : Central supplies a mix of BFG and COG, and Anchor BOS and COG. (The gas produced by Basic Oxygen Steel-making is termed BOS.) These constituent gases can themselves vary in content and quality. An indication of the majority supplier may be obtained by examining the CO content; e.g. a high percentage indicates that Anchor is the main supplier. [Table 6.1]

Table 6.1 (below) shows 'snapshots' of gas contents at low and high CO values with the right-hand two columns containing typical values for the two mixing stations [Appendix A1]. The figures for the two different CO levels would appear to broadly agree with the ranges for the two different mixing stations confirming CO as an indicator.

		HSM supply		Central	Anchor
Gas		At min CO (time 23475 mins)	At max CO (time 38004 mins)		
Carbon monoxide	CO	13.32	67.48	14.74	60.78
Carbon dioxide	CO_2	12.72	10.76	13.41	12.63
Hydrogen	H_2	29.7	5.45	28.88	9.17
Nitrogen	N_2	28.88	11.73	29.10	13.23
Methane	CH_4	13.37	3.61	11.5	3.108
Calorific values (MJ/Nm^3)		10.6	10.61	10.06	10.14
Specific gravity		0.71	0.96	0.738	0.945

Table 6.1 HSM fuel properties compared with typical values from both mixing stations

6.1 Analysis of data

Some 40, 000 data points were supplied covering a period of approximately 30 days during April/May 2000. Table 6.2 shows the ranges for the required, predicted parameters

Quantity / units	*Range*	*Mean*	*median*	*s.d.*
calorific value (MJ/Nm3)	7.93 – 11.47	9.75	9.62	0.48
air/fuel ratio	1.76 – 2.57	2.18	2.19	0.11
specific gravity	0.70 – 0.98	0.84	0.84	0.06

Table 6.2 Properties of parameters under investigation, from supplied data

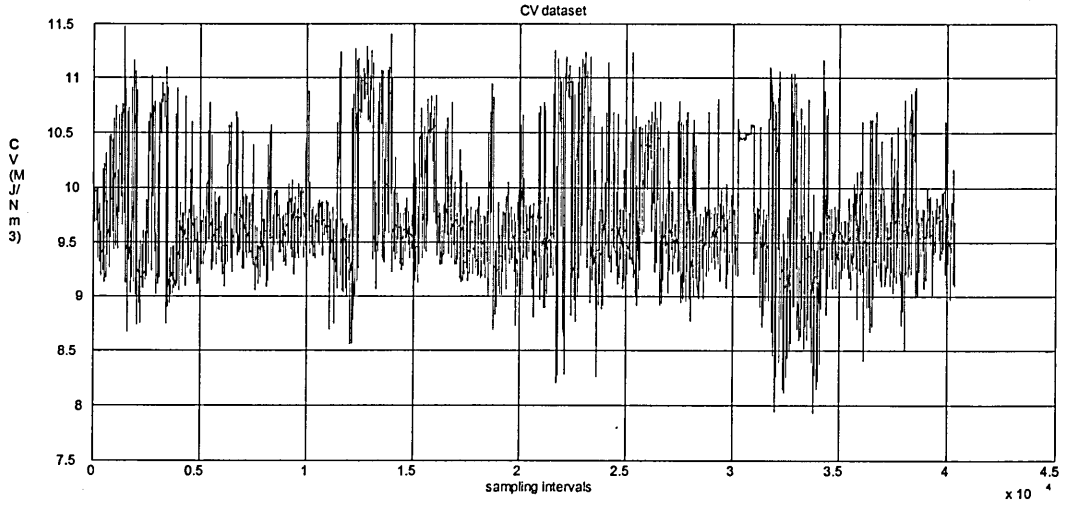


fig. 6.2 Variation in CV

Figure 6.2 shows the entire CV data set with its fluctuations, whilst figures 6.3 - 5 show a 'snapshot' of CV, CO and H_2 for the period 1,000 to 14,500 mins. From 6.3 it will be seen that there are clearly defined (often abrupt) swings between two discernible levels.

Examination of the CO plot suggests that gas supply is alternating between the two mixing stations.

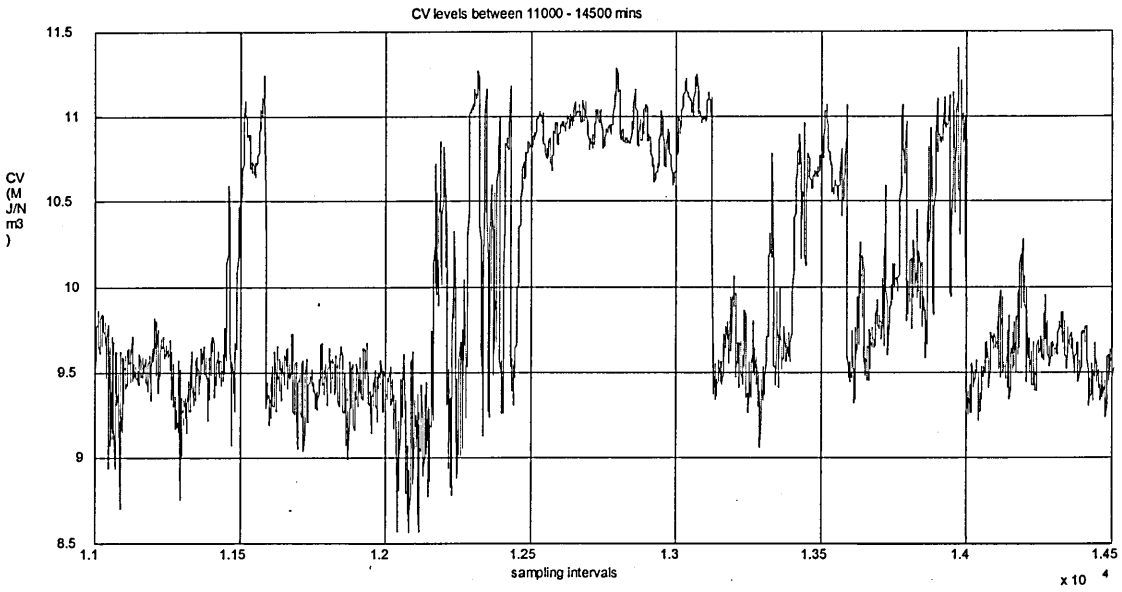


fig. 6.3 CV between 11000 and 14500 mins

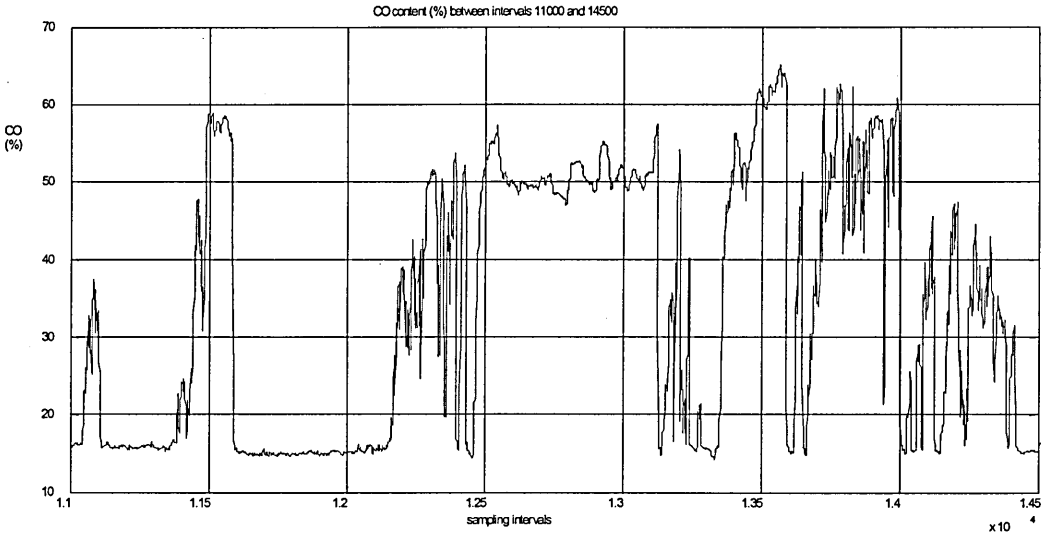


fig. 6.4 Carbon monoxide (%) for the above period

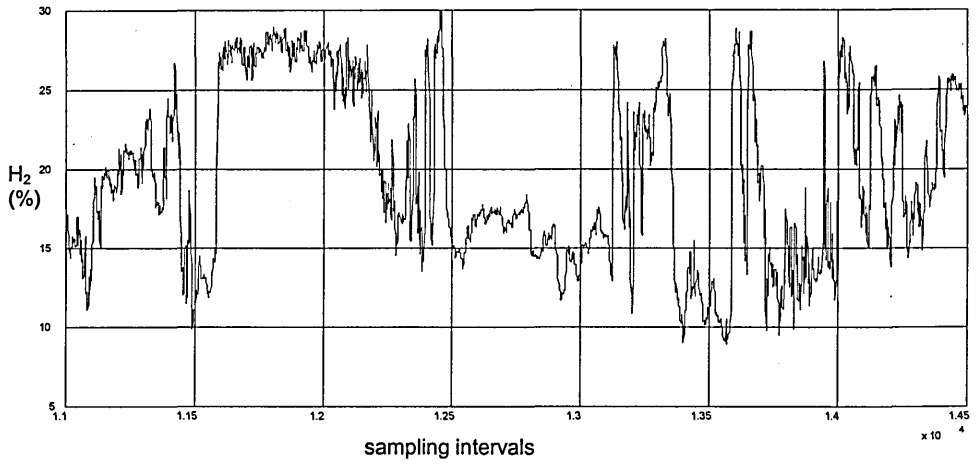


fig. 6.5 Hydrogen (%) for the same period

However whilst the CO plot is fairly steady in the first section of the graph, that of CV is still fluctuating. Figure 6.5 suggests that this is due to variations in the H_2 content. Given that at least some of the unpredictability was caused by the presence of two mixing stations supplying different forms of MEG in differing proportions and at varying rates, it was decided to examine the effect of introducing additional parameters (CO and H_2) as the network inputs which would function as an indicator to which mixing station was the dominant supplier, if any, at a given instant. [Section 6.3.3]

There was also a requirement for additional predicted parameters – air/fuel ratio (a/f), and specific gravity (s.g.). Additionally, since some of the data supplied for Phase 2 of the work was in the form of Wobbe numbers or indices (defined in Section 1), data for this parameter was synthesised using equation 1.1 and networks developed for comparison purposes.

Tables 6.3 to 6.6 and figures 6.6 to 6.8 show the variations in values for the above parameters.

set	*time (mins)*	*range*	*mean*	*median*	*s.d.*
training	15001 – 30000	8.2 – 11.25	9.79	9.63	0.48
validation	31000 – 40319	7.93 – 11.18	9.59	9.54	0.42
verification	1 – 15000	8.56 – 11.47	9.78	9.64	0.48

Table 6.3 CV development dataset characteristics

set	*time (mins)*	*range*	*mean*	*median*	*s.d.*
training	15001 – 30000	0.71 – 0.97	0.86	0.86	0.05
validation	31000 – 40319	0.70 – 0.98	0.84	0.84	0.06
verification	1 – 15000	0.71 – 0.96	0.82	0.82	0.005

Table 6.4 s.g. dataset characteristics

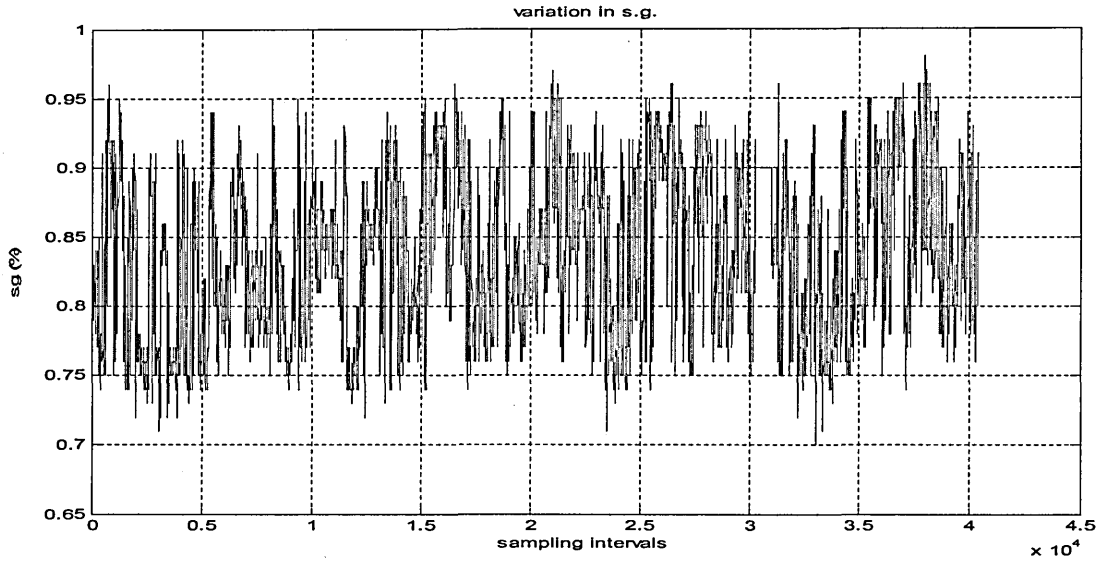


fig. 6.6 Variations in s.g. over entire data set

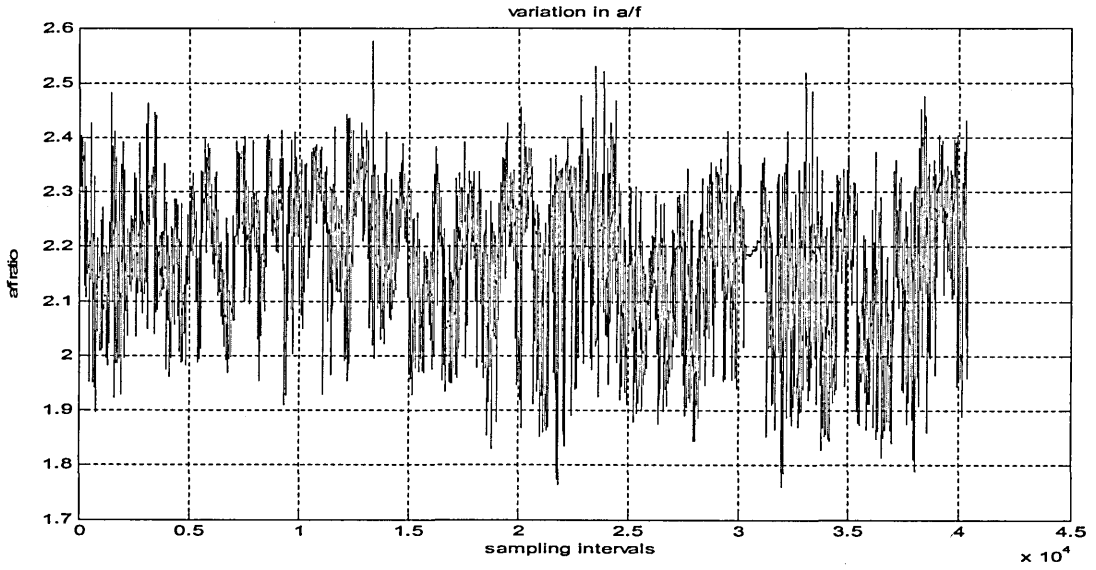


fig. 6.7 Variations in a/f ratio over entire data set

set	*time (mins)*	*range*	*mean*	*Median*	*s.d.*
training	15001 – 30000	1.76 – 2.54	2.16	2.17	0.11
validation	1 – 15000	1.90 – 2.58	2.21	2.22	0.10
verification	31000 – 40319	1.76 – 2.52	2.15	2.16	0.12

Table 6.5 a/f data characteristics (Note : different set groupings to above)

set	*time (mins)*	*range*	*mean*	*median*	*s.d.*
verification	15001 – 30000	8.72 – 12.58	10.59	10.54	0.47
training	31000 – 40319	8.51 – 12.68	10.46	10.46	0.44
validation	1 – 15000	9.07 – 12.74	10.78	10.70	0.44

Table 6.6 Synthesised Wobbe index datasets

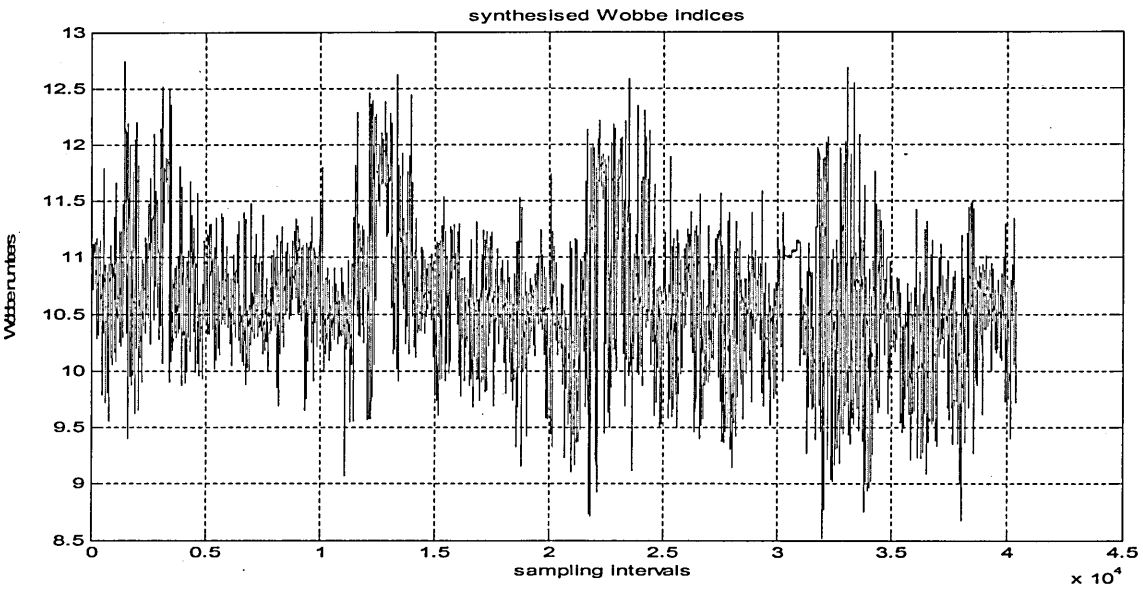


fig. 6.8 Synthesised Wobbe numbers

It should be noted that in the case of the datasets for air/fuel prediction the validation and verification sets have been interchanged. The set ranging from 1 - 15000 mins has somewhat different characteristics to the other two. Intuitively, it was felt that the verification set should reflect the dynamic characteristics of the training set.

6.1.1 Effect of pre-processing by 'm = 2'

Table 6.7 and figure 6.9 show the effects of pre-processing by the *'m = 2'* method on the datasets whilst table 6.8 provides the corresponding CPS datasets for comparison purposes.

set	*time (mins)*	*Range*	*mean*	*median*	*s.d.*
training	15001 – 30000	0.94 – 1.04	1.0	1.0	0.010
validation	31000 – 40319	0.93 – 104	1.0	1.0	0.010
verification	1 – 15000	0.95 – 1.03	1.0	1.0	0.005

Table 6.7 CV data characteristics after pre-processing by *m* = 2

dataset	*Range*	*mean*	*Median*	*s. d.*
Training	0.99 – 1.02	1.0	1.0	0.003
Validation	0.99 – 1.01	1.0	1.0	0.003
Verification	0.99 – 1.02	1.0	1.0	0.002

Table 6.8 equivalent CPS CV sets

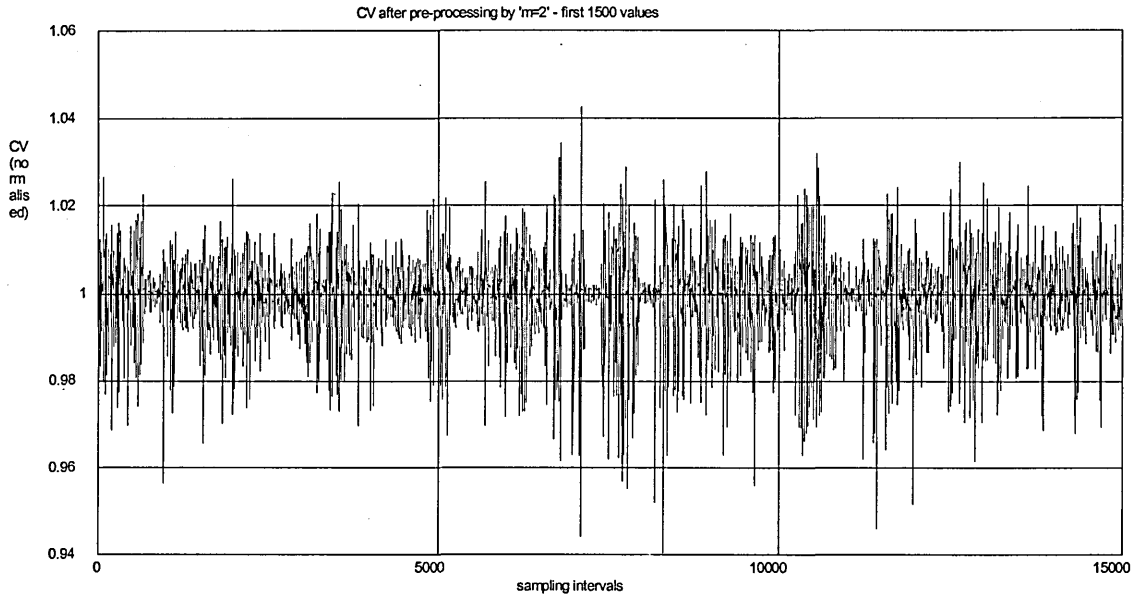


fig 6.9 pre-processing by $m = 2$ on CV (first 15000 mins)

It will be seen that as in the case of the CPS (table 6.8), all data sets are normalised about similar ranges. However the CPS sets are smaller, consisting only of several hundred data points, and of fewer constituent gases.

6.1.2 The possibility of a mass spectrometer failure

Whilst examining the data for potential training, validation and verification sets it was observed that the data exhibited some unusually near constant levels between approximately 30200 and 30980 mins (figs 6.10 to 6.13). (A duration of some 13hrs.)

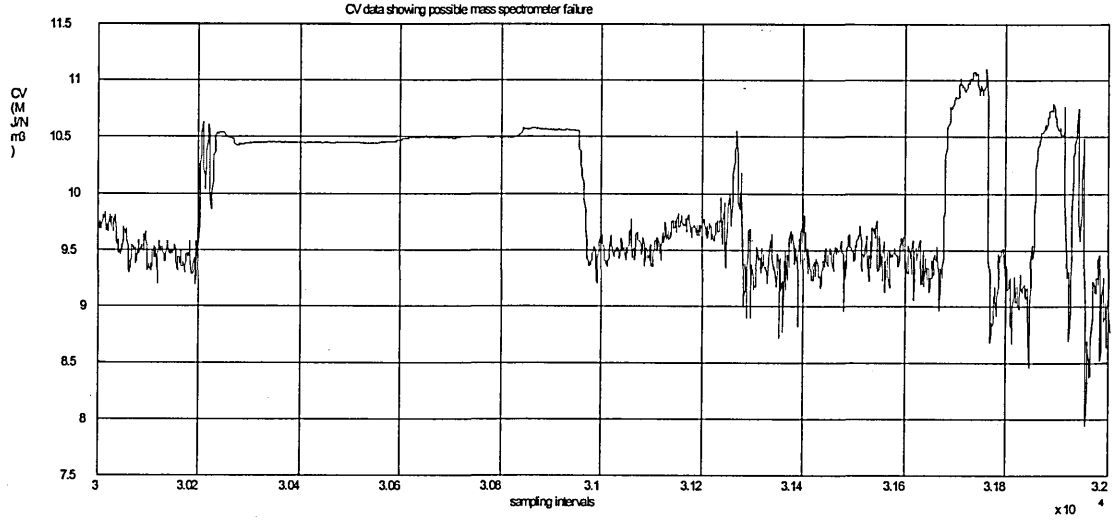


fig. 6.10 Possible mass spectrometer fault (CV data)

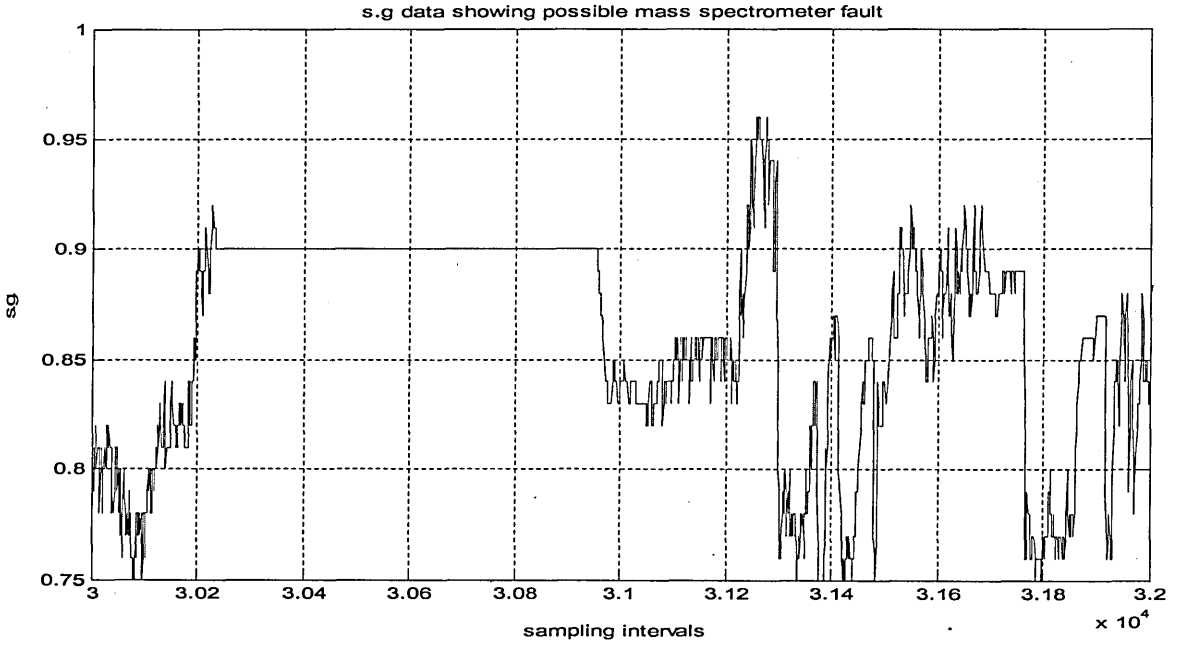


fig. 6.11 Possible mass spectrometer fault (s.g. data)

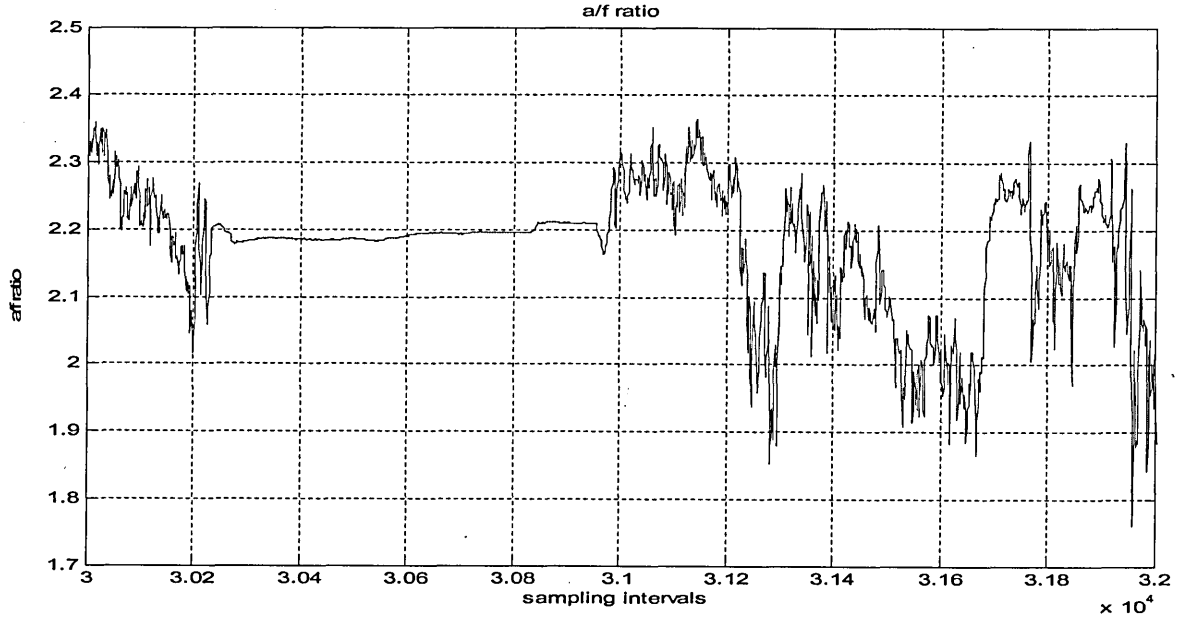


fig. 6.12 Possible mass spectrometer fault (a/f data)

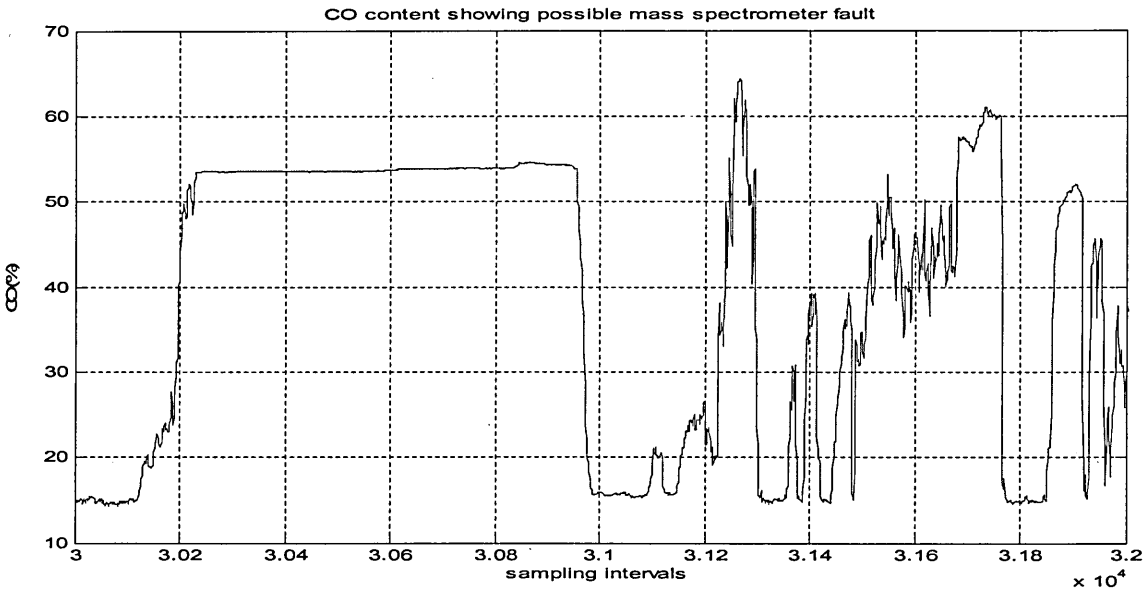


fig. 6.13 Percentage CO showing possible mass spectrometer fault.

Examination of the of the above graphs (figs 6.11 to 6.13) shows the s.g. plot to be constant whilst a/f ratio and CO content have a slow, minor upward trend, but devoid of the usual noise and abrupt excursions. In the absence of any information it is assumed that there is a fault condition and the above datasets for network development do not utilise data from this time period. However, the apparent fault condition for the CPS work (Table 5.1, page 32) indicates a zero level for c.v., whereas the above data shows a slow upward drift from 10.5 MJ/Nm^3 .

6.1.3 Revised experimental strategy

Informal preliminary testing (using both linear and non-linear networks) performed before substantive information was received on the nature and content of the HSM gas supply, and before adequate computing facilities became available, had produced extremely unpredictable results with error magnitudes of up to 35%. This was particularly true of non-linear network development which also witnessed regular instability induced within the training software. Hence it was decided not to proceed with any non-linear investigation for these datasets.

Further, it was suspected that the system dynamics were such that the range and frequency of the changes were not being fully captured by the mass spectrometer and that this caused a

further layer of aliasing in addition to that induced by the main data logging system operating with a much longer sampling interval than the mass spectrometer. Thus the 'm=2' method of pre/post-processing used with some success in the CPS and Dawes Lane work might only exacerbate the situation by introducing data smoothing with consequent loss of available dynamic knowledge with consequent reductions in accuracy. Analysis of the results for CV prediction (section 6.3.1) showed that to be the case and the 'm=2' method of pre/post-processing discarded for the remaining HSM parameter investigations.

6.2 Least-squares-based method

Information was received from Corus that a least-squares method had been investigated, using a second order equation with coefficients obtained from the previous 5 values to predict the next output. This had been performed on '1-minute' CV data from the HSM and found to be satisfactory. Thus an opportunity arose to obtain benchmarks for accuracy and an appropriate script file was implemented employing the Matlab 'polyfit' function.

Whilst Corus did not state that other parameters had been investigated, nor that more than one sampling interval ahead had been predicted, it was decided to also test the method in these circumstances; results are shown in table 6.9.

	1 sampling interval ahead			*2 sampling intervals ahead*		
parameter	***Mean***	***max***	***s.d.***	***mean***	***max***	***s.d.***
CV	1.31	18.13	1.72	2.85	43.24	3.65
s.g.	1.26	18.99	1.38	2.62	39.29	2.87
a/f	1.44	17.98	1.72	3.15	42.49	3.68
Wobbe index*	1.53	20.52	1.87	3.28	49.64	3.98

*synthesised data

Table 6.9 Results from least squares-based predictor

The actual results for Corus' investigation were not known but the result for predicting a single sampling interval ahead for CV appeared to infer that maximum absolute error values of up to 20% could be acceptable.

6.3 Summary of results and discussion

6.3.1 Calorific Value (CV)

Table 6.10 shows the maximum error range (percentage) for five approaches for one interval and two intervals ahead.

p.p./ method	max error range (%)	
	1 interval ahead	2 intervals ahead
Lsq*	18.13	43.24
None	11.40 – 11.43	16.78 – 16.99
÷100	11.40 – 11.44	16.78 – 16.99
$m = 2$**	12.27 – 12.34	17.74 – 17.76
'CV + CO' ^{m}	11.42 – 11.43	—

*least-squares approach ** after post-processing
^{m} multiple inputs of CV and CO (or H_2)

Table 6.10 CV maximum error ranges

Table 6.11 shows the mean, maximum and standard deviation in error (percentage) for five approaches for one interval and two intervals ahead.

p.p/ method	1 interval ahead					2 intervals ahead				
	struct	mean	max	s.d.	<0.01	struct	mean	max	s.d.	<0.01
lsq	—	1.31	18.13	1.72	—	—	2.85	43.24	3.65	—
none	3+K	0.57	11.40	0.71	0	3+K	0.97	16.78	1.11	0
÷ 100	3+K	0.57	11.40	0.71	0	3+K	0.97	16.78	1.11	0
$m = 2$**	4+K	0.70	12.27	0.86	1	3+K	1.04	17.74	1.26	0
CV + CO*	(3+2)+K	0.57	11.42	0.70	0	—	—	—	—	—

**after post-processing *results for cv +H_2 identical

Table 6.11 CV possible network solutions

The '$m = 2$' method consistently presents lower accuracy than no-preprocessing and the 'division by 100' methods. The least-squares approximation offers the poorest performance with an increase over the most accurate of solutions in maximum error of some 60% for predicting one sampling interval ahead , and 158% for two intervals ahead. The respective figures for mean error are 130% and 194%, while those for s.d. (standard deviation) are 149% and 229%.

Introducing a second input parameter of percentage carbon monoxide (or hydrogen) content results in a network with 3 CV inputs and 2 for CO (or H) (tables 6.10 and 6.11). The results are almost identical to those of the network trained with unprocessed data, and presents little justification for the increased demand on computing resources that the additional input requires.

Further, the above results demonstrate that scaling the training data through division by 100 does not enhance accuracy and again does not justify the increased computational demands.

6.3.2 Specific gravity (s.g.)

No pre-processing was required for this parameter since all values were < 1.0. Given the low accuracy figures in the investigations above, the '$m = 2$' method was discarded for this section and those that follow. Table 6.12 shows the maximum error range (percentage) for one interval and two intervals ahead.

max error range (%)	
1 interval ahead	*2 intervals ahead*
18.06 – 18.07	19.47 – 19.50

Table 6.12 s.g. maximum error ranges

Table 6.13 shows the error characteristics for a possible solution and compares it with the corresponding least squares simulation.

1 interval ahead					*2 intervals ahead*				
struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
lsq	1.26	18.99	1.38	—	lsq	2.62	39.29	2.87	—
7+K	0.81	18.06	0.81	1	6+K	0.96	19.47	1.17	0

lsq = 'least squares' simulation

Table 6.13 s.g. possible network solutions

Whilst all error parameters are higher in value than those for the most effective CV solutions, the network is absorbing information across a greater time span, 6 and 7 minutes, as opposed to some 3 minutes. The maximum error for one interval ahead was similar to the benchmark 'least squares' result, but that for two intervals ahead was considerably more accurate than the benchmark.

6.3.3 Air/fuel ratio (a/f)

Table 6.14 shows the maximum error range (percentage) for one interval and two intervals ahead. Again, there is little significant difference between scaling through division by 100 and the networks trained with unprocessed data.

max error range (%)	
1 interval ahead	*2 intervals ahead*
13.08 – 13.14	18.88 – 19.00

Table 6.14 a/f maximum error ranges

Table 6.15 shows the error characteristics for a possible solution and compares it with the corresponding least squares simulation.

1 interval ahead					*2 intervals ahead*				
struct	***mean***	***max***	***s.d.***	***<0.01***	***Struct***	***mean***	***max***	***s.d.***	***<0.01***
lsq	1.44	17.98	1.72	—	lsq	3.15	42.49	3.68	—
4+K	0.97	13.07	1.11	1	5+K	1.80	18.88	1.87	0

lsq = 'least squares' simulation

Table 6.15 a/f possible network solutions

The results do not achieve error levels comparable to those for CV; however the levels are lower than those for s.g., although requiring fewer inputs, i.e. less knowledge of the history of the data. The 'least squares' benchmarking results in significantly higher error levels than the neural network approach.

6.3.4 Wobbe number, or index

Table 6.16 shows the maximum error range (percentage) for one interval and two intervals ahead. Work with this parameter also demonstrated that scaling contributes little to training efficiency.

max error range (%)	
1 interval ahead	***2 intervals ahead***
13.08 – 13.14	18.88 – 19.00

Table 6.16 Wobbe index maximum error ranges

1 interval ahead					*2 intervals ahead*				
struct	***mean***	***max***	***s.d.***	***<0.01***	***struct***	***mean***	***max***	***s.d.***	***<0.01***
lsq	1.53	20.52	1.87	—	lsq	3.28	49.64	3.98	—
4+K	0.97	13.07	1.11	1	5+K	1.80	18.88	1.87	0

lsq = 'least squares' simulation

Table 6.17 Wobbe index possible network solutions

Table 6.17 shows the error characteristics for a possible solution and compares it with the corresponding least squares simulation. Contrary to the results obtained in Chapter 8 where predicting Wobbe index was found to be more accurate than predicting CV, table 6.17 shows an increase in error parameters when compared with the CV results above for 1 interval ahead (table 6.11). However the maximum absolute error for 2 intervals ahead shows a 17% improvement over that for CV. Nonetheless the other parameters remain higher. (It should be noted that the Wobbe data was synthesised from CV data sets.) With reference to table 6.9 the 'least squares' approach when applied to Wobbe indices produced the least accurate results.

6.3.5 Discussion

It was suspected that the data is changing too rapidly for a sampling interval of 1 minute in light of the results obtained in Chapters 7 and 8 with shorter sampling intervals. The '$m = 2$' method of pre-processing consistently exhibits lower accuracy, while scaling offers no benefits over networks trained without pre-processing of data.

Work with the least-squares-based method resulted in maximum absolute errors of up to 20% for one interval ahead, and approximately twice that when predicting two intervals ahead.

Further discussion and comparison of the above results is included at the end of Chapter 8 (HSM Phase2).

7.0 Dawes Lane Coke Ovens

There are two sets of coke ovens (Dawes Lane and Appleby) supplying the Scunthorpe site with COG. This investigation considers data from the Dawes Lane plant. Of particular significance in this data was the 23 sec. sampling interval. This meant that the data was obtained directly from the mass spectrometer output and therefore not subjected to aliasing through being only logged at 1 min intervals by the main logging system as in all previous data. Thus there was an opportunity to compare the effect of '1 minute' logging with actual mass spectrometer output by artificially generating one-minute data.

7.1 Analysis of data

percentage constituents		*range*	*mean*	*median*	*std. dev.*
Carbon monoxide	CO	5.59 - 6.22	5.94	—	0.09
Carbon dioxide	CO_2	1.09 - 1.49	1.25	—	0.08
Hydrogen	H_2	62.26 – 67.89	65.53	—	1.17
Nitrogen	N_2	0.70 - 3.68	1.52	—	0.68
Methane	CH_4	21.02 – 24.25	22.73	—	0.75
Ethylene	C_2H_4	1.65 – 2.00	1.84	—	0.08
Ethane	C_2H_6	0.55 - 0.71	0.64	—	0.04
Calorific value (MJ/Nm^3)		16.99 – 18.41	17.81	17.84	0.28
Specific gravity		0.28 – 0.32	0.30	0.30	0.01
air/fuel ratio (true)		4.11 – 4.49	4.33	4.33	0.08

Table 7.1 COG characteristics (March 12th onwards)

Percentage constituents		*range*	*mean*	*meridian*	*std. dev.*
Carbon monoxide	CO	5.64 – 6.43	5.82	—	0.11
Carbon dioxide	CO_2	1.25 - 1.47	1.33	—	0.05
Hydrogen	H_2	63.25 - 66.76	65.33	—	0.81
Nitrogen	N_2	0.69 - 3.59	1.48	—	0.71
Methane	CH_4	21.69 – 24.00	22.93	—	0.50
Ethylene	C_2H_4	1.74 - 1.95	1.85	—	0.05
Ethane	C_2H_6	0.61 - 0.73	0.67	—	0.02
Calorific value (MJ/Nm^3)		17.29 – 18.35	17.91	17.93	0.21
Specific gravity		0.29 – 0.32	0.30	0.30	0.01
air/fuel ratio (true)		4.19 - 4.47	4.36	4.36	0.06

Table 7.2 COG characteristics (March 15th onwards)

There are two files covering a 24-hour period on 12th March 2001 (Table 7.1) and a 9-hour period from midnight to 9.30am on 15th March 2001 (Table 7.2). (The magnitudes are similar to those given in appendix A1 for typical COG values.) Given that the larger file had the wider range of dynamics it was decided to select sets from it for training and validation, and to verify against the March 15th file.

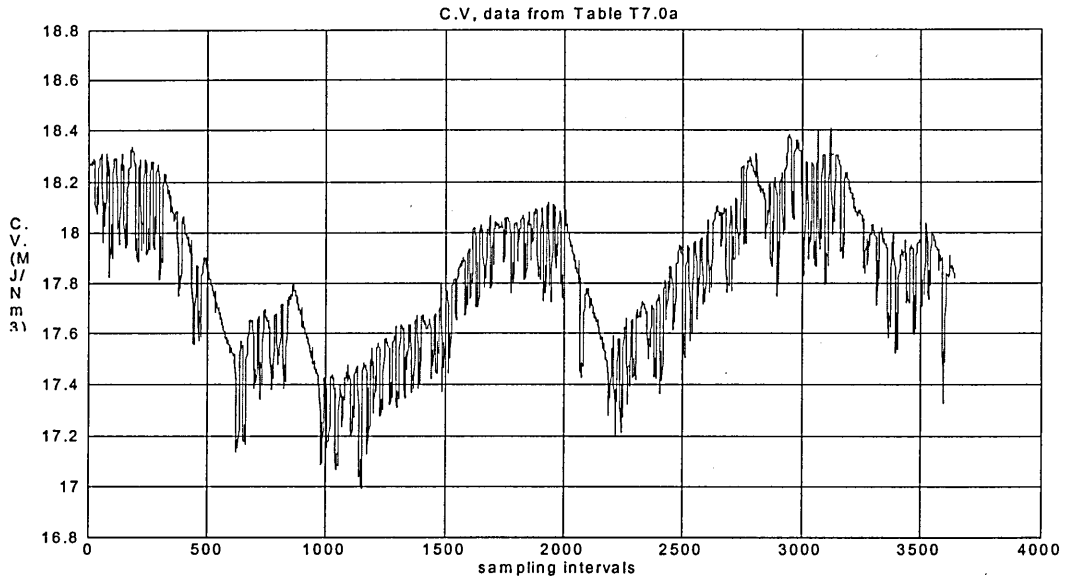


fig. 7.1 CV March 12th onwards

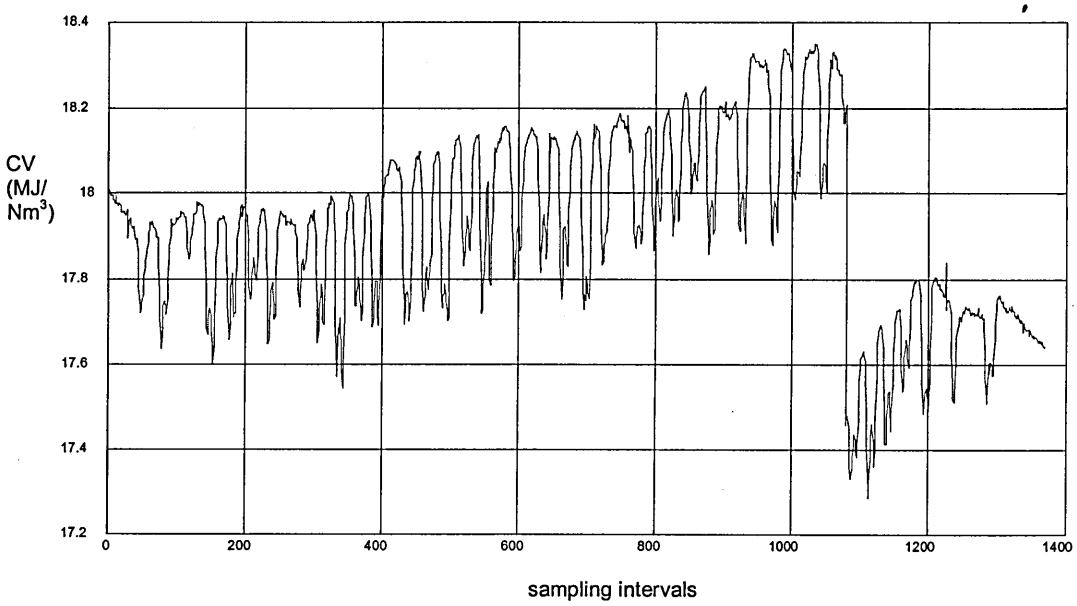


fig. 7.2 CV March 15th onwards

Fig. 7.1 shows a slow overall undulation with what appear to be noise spikes. Closer examination of the first 250 points (fig. 7.3) reveals that the spikes are in fact slow moving, and, when they do occur are of approximately 26 mins duration falling by around 400 kJ/Nm^3.

It is assumed in the absence of confirmation from Corus, that these variations illustrate fluctuations in the quality and type of fuel used.

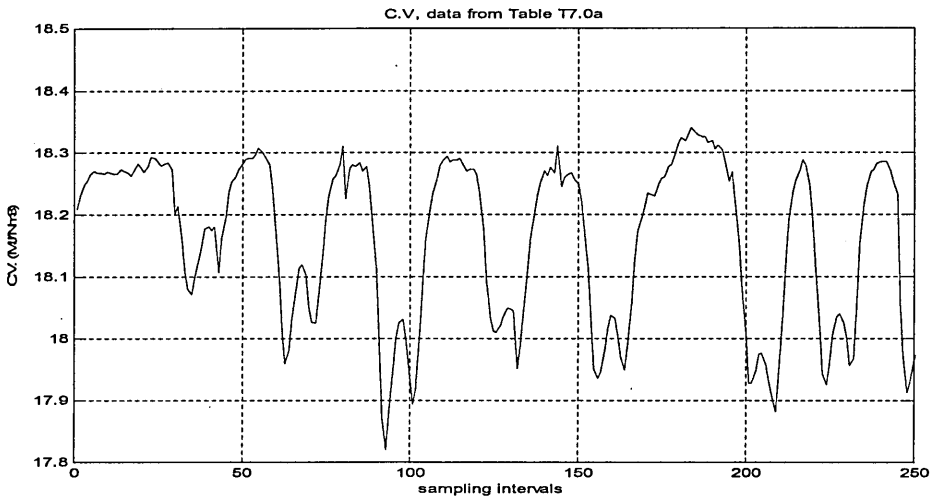


fig. 7.3 First 250 data points from fig. 7.1

Note that in fig. 7.2 there is a major change in level between sampling intervals1080 and 1081 from 18.2 to 17.46 MJ/Nm^3, i.e. a gradient of 0.74 MJ/Nm^3 per min. The maximum gradient encountered in the training data was –0.1762 MJ/Nm^3 per min, located between intervals 245 and 246 secs. Unlike the HSM data, these follow broadly the same trend as a result of only a single gas type being present.

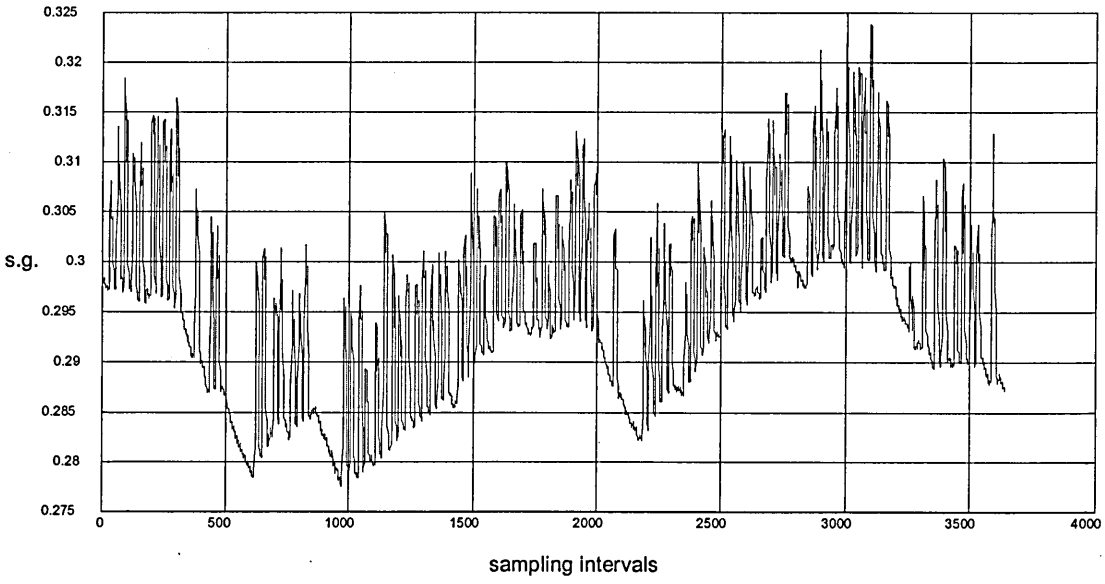


fig. 7.4 s.g. March 12th onwards

Figure 7.4 shows the s.g. data for March 12th onwards. Note that the fluctuations in level occur in the opposite direction to those of CV (fig.7.1).

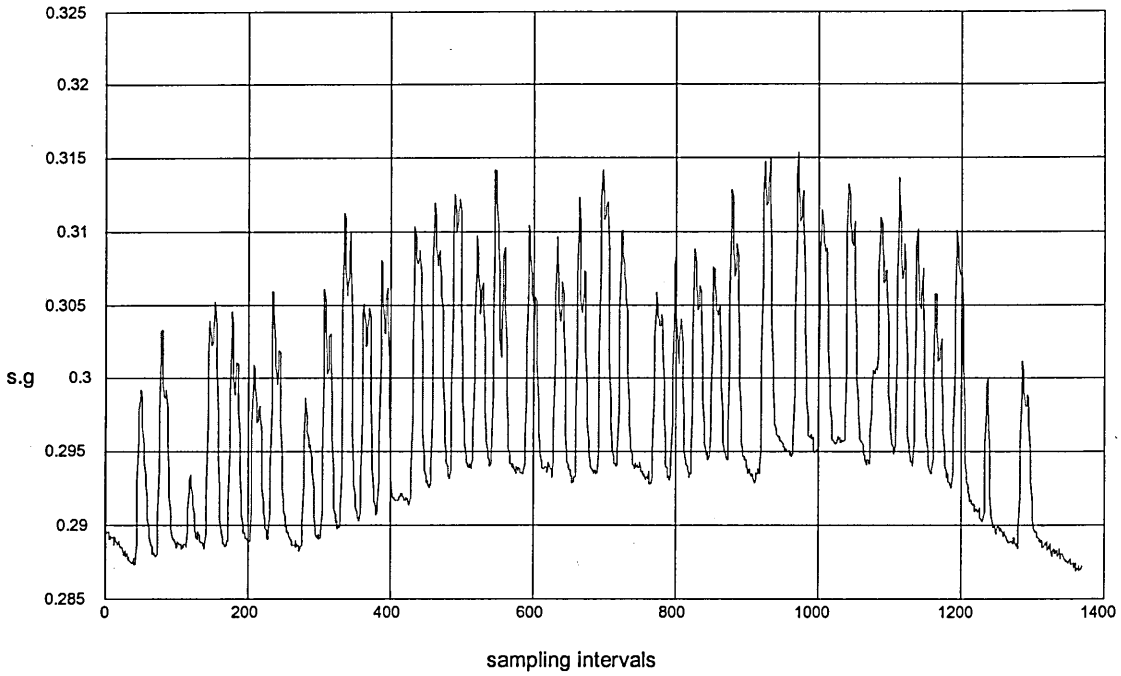


fig. 7.5 s.g. March 15th onwards

Figure 7.5 shows the s.g. corresponding to that shown in fig. 7.2; there is no corresponding abrupt change in levels at sampling interval 1080 and the data remains constant there at 0.3.

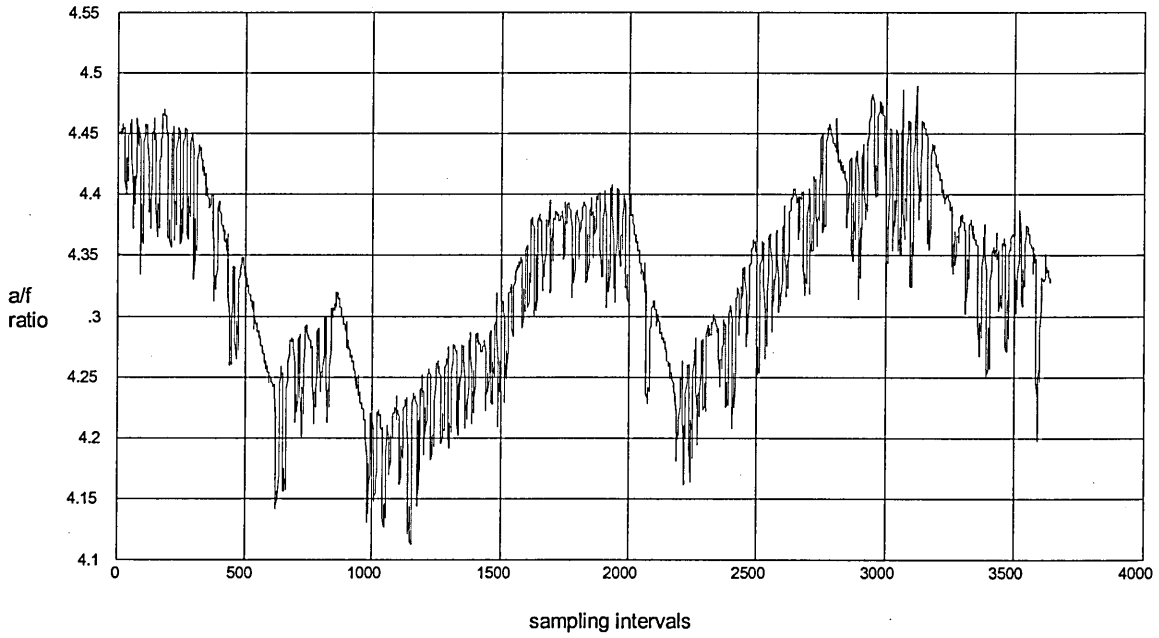


fig. 7.6 air/fuel ratio March 12th onwards

Comparison of fig.7.6 with fig. 7.1 and of fig. 7.7 below with fig. 7.2 above suggests that the dynamics of the air/fuel data reflect the changes in CV.

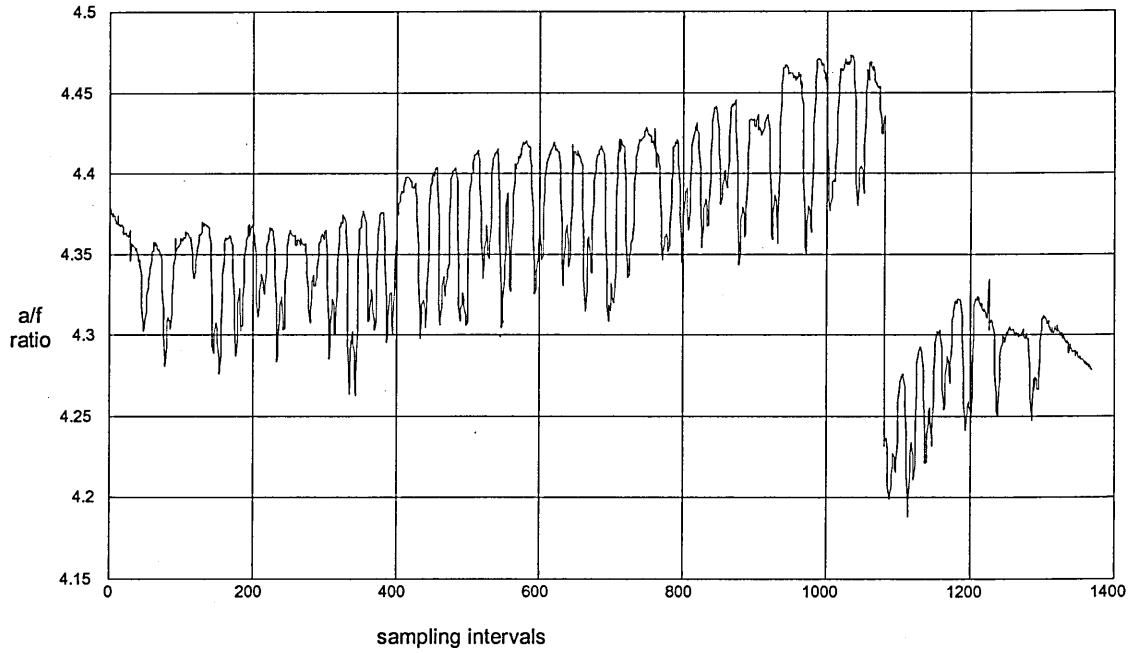


fig. 7.7 air/fuel ratio March 15th onwards

The methods established during the CPS investigations were used to predict values of CV, specific gravity (s.g.) and air/fuel ratio (a/f) using both linear and non-linear networks, and for predictions of up to two sampling intervals ahead. Further, data with one-minute sampling intervals was synthesised in order to validate the results obtained in the CPS section with longer data sets, and to examine the effect of the longer sampling interval on accuracy, i.e. the effect of introducing aliasing.

(Although work in chapter 6 with fast-changing gas content datasets had revealed a tendency for the 'm=2' pre-processing method to produce higher error levels, it was decided to continue with that method for the slower-changing Dawes Lane datasets.)

Data sampled at 1-minute intervals was synthesised from both supplied 23-sec data sets. (Section 7.0). The results are shown in figures 7.8 and 7.9.

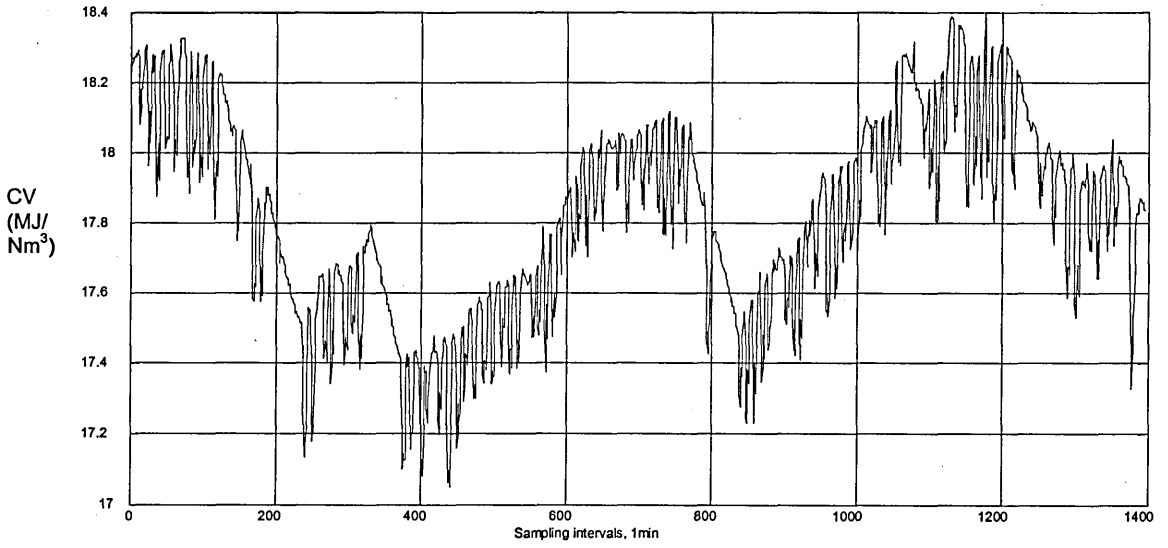


fig. 7.8 CV (1min.) March 12th onwards

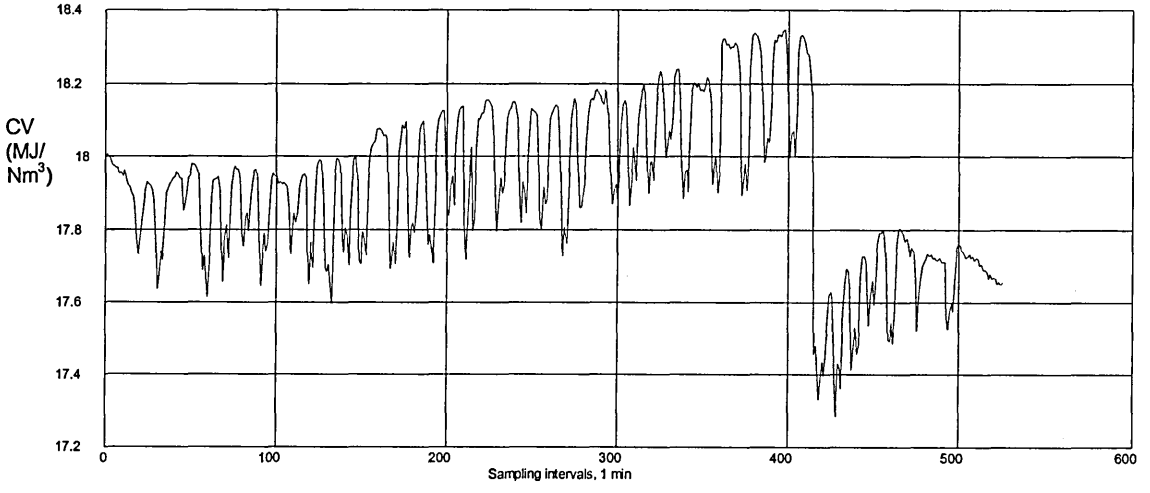


fig. 7.9 CV (1min.) March 15th onwards

dataset	*1 min*				*23 sec*			
	range	*mean*	*median*	*s.d.*	*Range*	*mean*	*median*	*s.d.*
Mar 12th	17.05 – 18.4	17.81	18.84	0.28	16.99 – 18.41	17.81	17.84	0.28
Mar 15th	17.29 – 18.35	17.91	17.93	0.22	17.29 – 18.35	17.91	17.93	0.21

Table 7.3 Comparison of CV data for 1min and 23 sec intervals

Table 7.3 compares the characteristics of the two types of dataset. Apart from a minor change in range, the differences between the two are not greatly significant.

dataset		1 min			
		Range	mean	median	s.d.
Dawes Lane	Mar 12th	17.05 – 18.4	17.81	18.84	0.28
	Mar 15th	17.29 – 18.35	17.91	17.93	0.22
CPS	training	18.03 – 22.09	20.65	20.81	0.94
	validation	17.42 - 19.87	18.59	18.55	0.56
	verification	17.07 – 20.29	19.28	19.49	0.81

Table 7.4 Comparison of 1min CV data for CPS and Dawes Lane

Table 7.4 compares the relevant values from the corresponding CPS datasets. These latter are higher in range and cover a broader span of values, with the higher standard deviation indicating a wider range of dynamics.

7.1.1 Effect of m = 2 pre-processing

The inherent post-processing effects on error are shown in table 7.5, whilst table 7.6 shows the changed data characteristics due to the method. A large spike is present in the verification set at interval 410 as a result. (Fig. 7.10)

Intervals Ahead	Mean	max	s.d.
1	0.0016	1.98	0.0024
2	0.004	0.0422	0.006

Table 7.5 error induced during post-processing

Data	range	span	mean	s.d.
Mar 12th	0.99 - 1.0091	0.0191	1.0	0.0022
Mar 15th	0.98 - 1.0079	0.0277	1.0	0.0024

Table 7.6 changed characteristics after pre-processing.

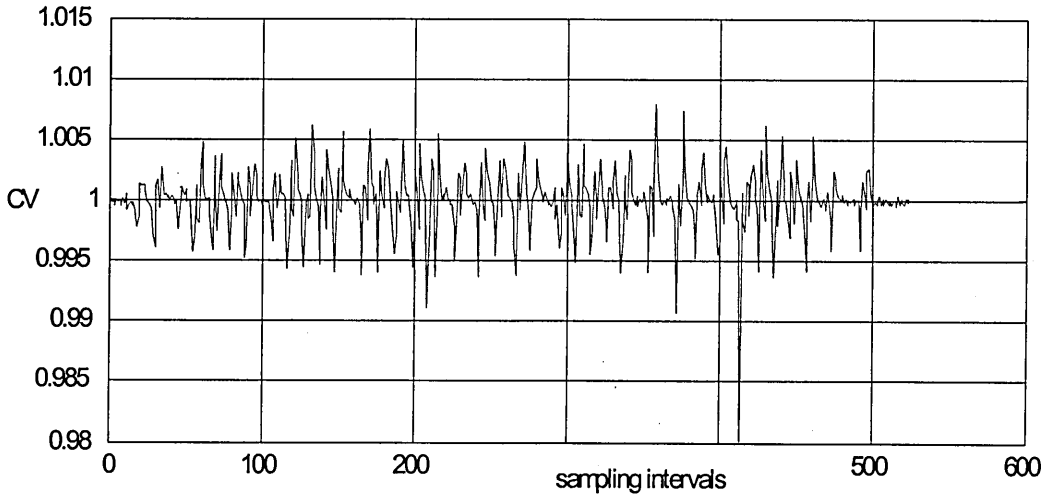


fig. 7.10 CV after pre-processing by '*m* = 2'

7.1.2 Possibility of mass spectrometer fault

Analysis of the results from the investigations using both 1-minute and 23-second sets of data showed the high maximum error to be located at a previously (in terms of training) unseen major abrupt change in values within the verification sets, at 1080 intervals for the 23-second set and 414 for the 1-minute data. Further analysis of the nearby data is illustrated in figure 7.11 and in more detail in 7.12.

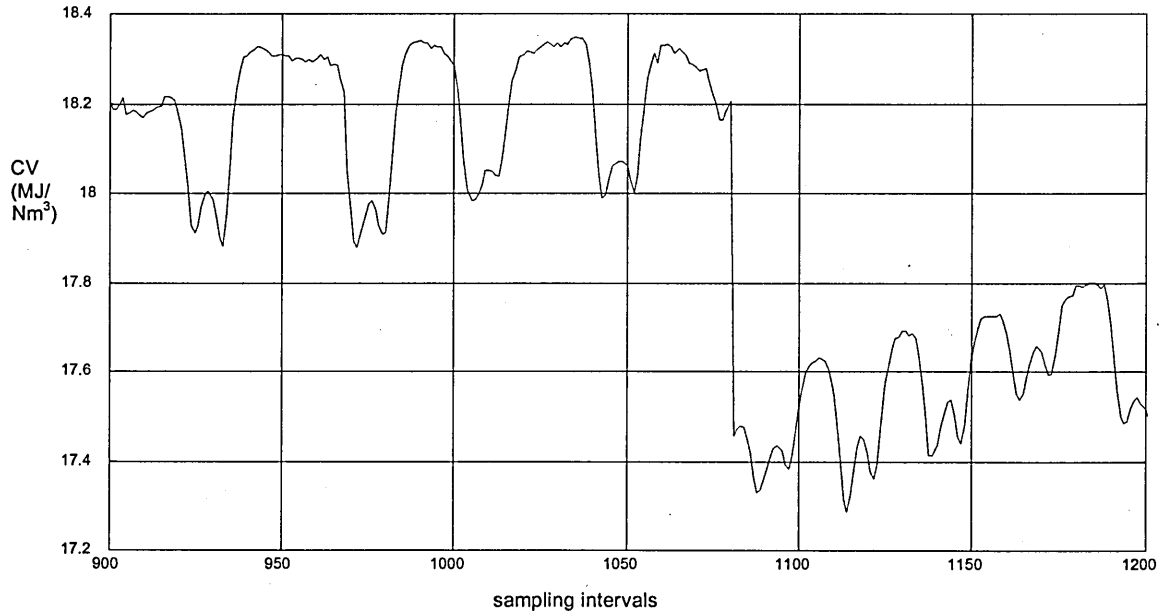


fig. 7.11 23-sec CV data near abrupt change

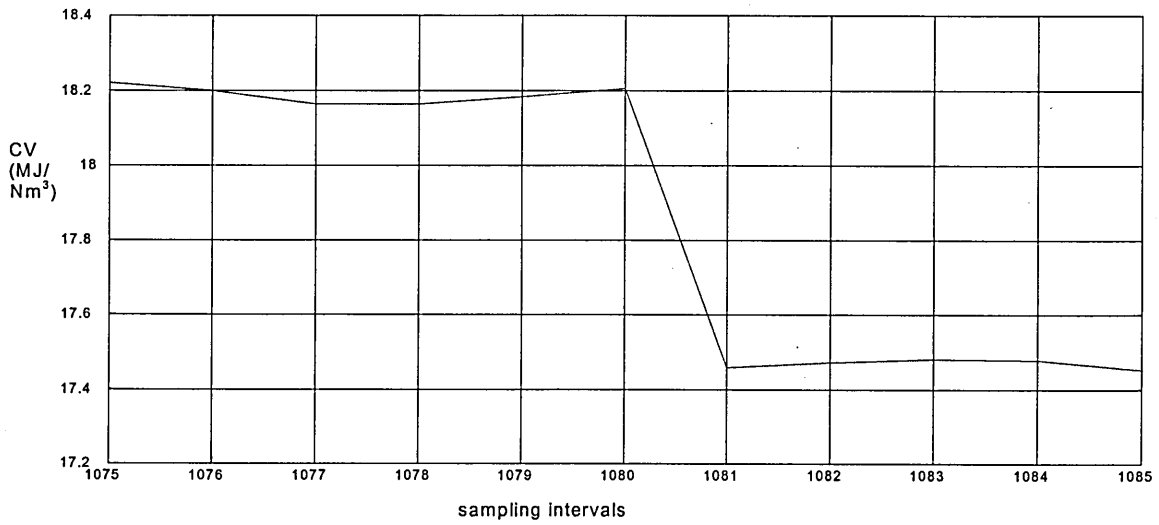


fig. 7.12 close-up of above

The CV levels either side of the change are 18.21 and 17.46 at intervals 1080 and 1081 respectively. The geometry of the graph suggests that but for the abrupt change in level the

curve would continue. Adding the difference in values (0.749 MJ/Nm^3) between the two points as an offset to the points from interval 1081 onwards produced the fig. 7.13.

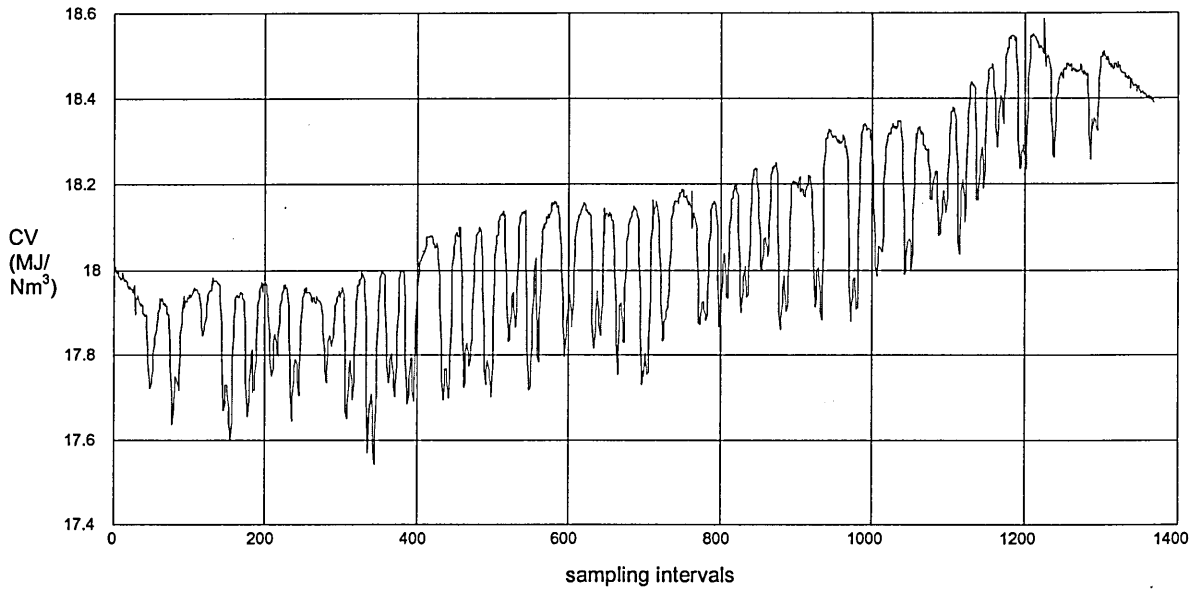


fig. 7.13 0.749 MJ/Nm^3 added to the latter part of the 23-sec data

Examination of fig. 7.2 shows that there is an approximate periodicity of 1500 – 2000 intervals per 'rise' cycle; the above alteration to the latter section of the date now appears to have brought it into line with the remainder of the data. Hence a tentative conclusion is that the above data set now matches this and thus suggests the possible existence of a mass spectrometer fault.

Further experiments were therefore carried out on the existing networks developed in the previous sections using the linear networks as indicators, in order to re-examine the results when verified without out the above gradient being present. Two verification sets were used : the first with data before the possible fault and the second with data after it to investigate how quickly a given network resumed efficient operation after encountering a possible fault condition. Both unprocessed data and data scaled through division by 100 were used; the '*m* = 2' method was not employed in view of the change in training data characteristics inherent in that method.

(Results are reported in Section A5.3 and commented on in Section 7.2.2.)

7.2 Summary of results and discussion

7.2.1 Calorific Value (CV)

p.p	Function	C.P.S	Dawes Lane	
			1-minute	23-second
none	Linear	1.52 – 1.79	3.91 – 4.15 (1.74 – 1.96)[1]	4.34 – 4.42 (0.83 – 0.88)[1]
	Log	1.32 – 20.5	3.29 – 4.31	4.33 – 4.73
	Tanh	1.30 – 21.4	3.88 – 35.63	3.27 – 4.41
÷100	Linear	1.49 – 1.71	3.91 – 4.15 (1.69 – 1.96)[1]	4.34 – 4.42 (0.83 – 0.88)[1]
	Log	1.33 – 37.5	3.28 – 4.51	3.80 – 4.38
	Tanh	1.57 – 10.1	3.62 – 6.70	3.75 – 9.09
$m = 2$**	Linear	2.11 – 2.19	4.24 – 4.32	4.27 – 4.29
	Log	2.07 – 4.43	4.14 – 6.90	4.27 – 751.72
	Tanh	1.99 – 4.43	4.08 – 4.94	4.27 – 67.39

** after post-processing [1] from section 7.3

Table 7.7 Maximum error ranges for CV, one interval ahead

Maximum absolute error ranges for both 'un-preprocessed' data and '÷100' are almost identical (table7.7). The '$m = 2$' method produces after post-processing, the lowest maximum errors for the linear solutions, although the mean error is some 60% higher with a greater errror standard deviation.

However, pre-processing by this method alters the dynamics of the of the training, validation and verification sets (e.g. Section 7.1.1); essentially the networks are being developed using different data to the other two methods of pre-processing with all data being distributed about an approximate mean of 1.0. As an example, the verification set contains a previously unseen abrupt change from 18.21 to 17.46 MJ/Nm^3 (-750kJ/ Nm^3 , or 4%), possibly due to a measurement malfunction, at interval 414 for the 1-minute data. (The effects of this are reported in more detail in Section 7.1.2) Figure 7.13 shows how the '$m = 2$ ' method has reduced this to a fall of approximately -0.02 (2%) at sampling minute 410. With the exception of non-linear network results for the 23-second data, the '$m = 2$' ranges remain broadly similar for all network types, although the lower limits tend to be higher than other forms of pre-processing, possibly indicating the resilience of this type of network to unforeseen major fluctuations in data sets not previously encountered during training.

From Table 7.7 the effect of increasing the number of points in each set by reducing the sampling interval to 23 seconds is that error ranges increase; the 23-second sets have most points and the CPS sets the least. However work in Section 7. 1.2 demonstrated that for the Dawes Lane sets this is due to the above gradient obscuring the improvement in accuracy by the introduction of a higher sampling frequency. In fact, there is a potential 50% reduction in maximum error when utilising the 23-second data, with corresponding fall in mean error and standard deviation of some $^2/_3$.

The methods employing non-pre-processed data and pre-processing through division by 100 produce near identical results, the differences being assumed to arise from different initial random weighting in the training method. However informal observation of the training times showed that there was a 30% reduction for the latter method.

With the exception of the CPS '÷100' networks, non-linear solutions have reduced the lower limits of the maximum error ranges, especially in the case of the tanh activation function, although that particular function resulted in many instances of singular matrix warnings and potential software instability during training.

Table 7.9 (next page) lists the most accurate networks (in terms of maximum error) for each method and activation function, together with acceptable computationally less-demanding solutions where different. The error ranges for those solutions are summarised in table 7.8 below.

Error	*CPS*	*Dawes Lane*	
		1-minute	*23-second*
mean	0.28 – 0.40	0.26 – 1.24	0.10 – 1.22
max	1.72 – 3.22	3.29 – 4.24	3.27 – 4.38
s.d.	0.25 – 0.37	0.32 – 0.89	0.17 – 0.87

Table 7.8 Comparison of error parameters for CV

The CPS results have the lowest maximum error ranges but development and verification involved fewer data points. There is little difference between the Dawes Lane 1-minute and 23-second results. Both verification sets contained a possible measurement malfunction. However the lower limits of the mean error and standard deviation are much reduced for the higher sampling frequency. With regard to Section 7.3 and Table A5.3.5, when verification was performed on the linear solutions with the malfunction omitted, a further improvement in maximum error by the 23-second set over the 1-minute set was observed.

In terms of the number of inputs, i.e. the number of previous samples required to predict one sampling interval ahead, the CPS solutions require 3 – 5 or 11 – 12 "past values" (inputs) . For the Dawes Lane '1-minute' networks either 4 – 6 or 15 – 17, and for the '23-second' data two networks require 16 past values and the remainder 3 – 6. However, in terms of accuracy the results appear evenly distributed between small and large networks.

	CPS					Dawes Lane 1-min					Dawes Lane 23-sec				
p/p	*funct*	*struct*	*mean*	*max*	*s.d.*	*funct*	*struct*	*mean*	*Max*	*s.d.*	*funct*	*struct*	*mean*	*max*	*s.d.*
lsq	—	—	0.29	1.87	0.25	—	—	0.59	3.61	0.57	—	—	0.15	4.60	0.24
none	linear	13+K	0.28	2.06	0.25	linear	3+K	0.30	3.93	0.34	linear	16+K	0.19	4.38	0.27
	log	11+13+K	0.31	1.72	0.26	log	5+2+K	1.24	3.29	0.89	log	3+2+K	0.1	4.33	0.17
		12+4+K	0.31	2.03	0.26		17+3+K	1.24	3.3	0.91	tanh	3+2+K	1.22	3.27	0.87
	tanh	12+6+K	32	1.72	0.27		14+2+K	0.99	3.74	0.72					
		12+3+K	0.31	1.88	0.27	tanh	4+2+K	0.29	3.91	0.34					
							17+4+K	0.26	3.88	0.33					
							17+2+K	0.28	3.92	0.32					
÷100	linear	4+K	0.21	1.39	0.20	linear	4+K	0.29	3.91	0.34	linear	16+K	0.19	4.38	0.27
		12+K	0.28	2.04	0.25	log	4+6+K	0.29	3.59	0.33	log	6+2+K	0.38	3.8	0.39
	log	12+7+K	0.4	1.95	0.31		6+2+K	0.28	3.75	0.37	tanh	6+2+K	0.34	3.75	0.30
		12+2+K	0.31	2.23	0.27		16+2+K	1.02	3.28	0.72					
	tanh	11+4+K	0.35	1.72	0.31	tanh	4+4+K	0.29	3.62	0.37					
		12+2+K	0.37	2.57	0.34		3+2+K	0.29	3.89	0.37					
							15+3+K	0.28	3.83	0.36					
							16+2+K	0.28	3.95	0.35					
m = 2	linear	5+K	0.37	3.22	0.33	linear	3+K	0.44	4.24	0.47	linear	3+K	0.17	4.25	0.21
	log	12+7+K	0.40	3.11	0.36	log	4+4+K	0.34	4.14	0.39	log	4+2+K	0.15	4.27	0.20
		12+3+K	0.37	3.16	0.34		3+2+K	0.36	4.20	0.41	tanh	3+3+K	0.15	4.27	0.20
	tanh	11+7+K	0.39	3.06	0.35		16+4+K	0.38	4.15	0.40		5+2+K	0.15	4.27	0.20
		12+3+K	0.4	3.22	0.37		14+2+K	0.35	4.19	0.39					
						tanh	5+5+K	0.33	4.15	0.37					
							5+2+K	0.35	4.20	0.44					
							17+3+K	0.34	4.08	0.38					
							14+2+K	0.39	4.22	0.45					

Table 7.9 CV solutions with suitable minimalist networks (1 interval ahead)

Of particular note is that several results show a fall in maximum absolute error at the expense of an increase in mean error to values greater than 1%, with corresponding increases in standard deviation. It is possible that some overtraining has taken place despite the presence of a validation set.

Tables 7.10 to 7.12 show the corresponding results for predictions two intervals ahead.

p.p	*function*	*C.P.S*	*Dawes Lane*	
			1-minute	*23-second*
None	linear	2.06 – 2.36	4.09 – 4.35	4.38 – 5.09
	log	1.72 – 20.5	3.29 – 19.14	4.24 – 46.61
	tanh	1.74 – 2.84	3.59 – 5.79	3.94 – 65.01
÷100	linear	1.97 – 2.30	4.10 – 4.35	4.38 – 5.11
	log	1.95 – 12.19	3.56 – 8.25	3.89 – 8.57
	tanh	1.72 – 75.9	3.57 – 11.83	4.02 – 6.46
m = 2**	linear	3.22 – 3.28	4.60 – 4.65	4.25 – 4.27
	log	3.11 – 3.52	4.47 – 39.14	4.17 – 67.4
	tanh	3.06 – 3.49	4.50 – 16.0	4.17 – 342.68

** after post-processing

Table 7.10 Maximum error ranges for CV, two intervals ahead

In table 7.10 the lower limits of the maximum error ranges show a slight increases over those in table 7.2. Again there is evidence of non-linear solutions producing high upper limits.

Error	*CPS*	*Dawes Lane*	
		1-minute	*23-second*
mean	0.28 – 0.40	0.48 – 1.24	0.17 – 1.01
max	1.72 – 3.22	3.29 – 4.54	3.89 – 4.38
s.d.	0.25 – 0.34	0.48 – 0.89	0.24 – 0.74

Table 7.11 Summary of ranges in 7.10

Again there are lower maximum error levels for the CPS, and similar ranges for Dawes Lane, both for 1-minute and 23-second data (tables 7.11 and 7.12). Also the 23-second data has lower ranges for mean error and standard deviation when compared to the 1-minute data. With respect to the latter there is no change in the maximum error lower limits between predicting 1 interval head and two, although the values for mean error and standard deviation are increased by some 50%. Of interest in both those cases is that the upper limits are unchanged. As in the '1 interval ahead' results, the 23-second data is capable of significant reductions in mean error and standard deviation.

	CPS					Dawes Lane 1-min					Dawes Lane 23-sec				
p/p	***funct***	***struct***	***mean***	***Max***	***s.d.***	***funct***	***Struct***	***mean***	***max***	***s.d.***	***struct***	***funct***	***mean***	***max***	***s.d.***
lsq	—	—	0.44	3.16	0.42	—	—	1.22	8.34	1.14	—	—	0.37	8.51	0.51
none	linear	13+K	0.28	2.06	0.25	linear	4+K	0.53	4.09	0.53	linear	16+K	0.19	4.38	0.27
	log	11+13+K	0.31	1.72	0.26	log	4+2+K	1.24	3.29	0.89	log	15+3+K	0.17	4.24	0.24
		12+4+K	0.31	2.03	0.26	tanh	5+2+K	0.48	3.59	0.48		17+2+K	0.17	4.33	0.27
	tanh	12+6+K	0.32	1.72	0.27						tanh	15+2+K	1.01	3.94	0.74
		12+3+K	0.31	1.88	0.27										
÷100	linear	3+K	0.29	1.97	0.26	linear	4+K	0.53	4.1	0.53	linear	16+K	0.19	4.38	0.27
	log	12+7+K	0.40	1.95	0.31	log	5+2+K	0.48	3.56	0.48	log	16+5+K	0.4	3.89	0.39
		12+2+K	0.31	2.23	0.27	tanh	5+5+K	0.48	3.57	0.48		17+3+K	0.17	4.26	0.28
	tanh	11+4+K	0.35	1.72	0.31						tanh	17+4+K	0.58	4.02	0.53
		12+2+K	0.37	2.57	0.34							17+2+K	0.17	4.27	0.27
m = 2	linear	5+K	0.37	3.22	0.33	linear	3+K	0.57	4.6	0.6	linear	3+K	0.17	4.25	0.21
	log	12+7+K	0.40	3.11	0.36	log	4+4+K	0.55	4.47	0.59	log	3+5+K	0.27	4.17	0.32
		12+3+K	0.37	3.16	0.34		6+2+K	0.57	4.54	0.59		5+2+K	0.29	4.17	0.34
	tanh	11+7+K	0.39	3.06	0.35	tanh	6+2+K	0.54	4.5	0.56	tanh	4+5+K	0.27	4.17	0.33
		12+3+K	0.40	3.22	0.37							3+2+K	0.27	4.20	0.32

Table 7.12 CV solutions with suitable minimalist networks (2 intervals ahead)

With the exception of the Dawes Lane '$m = 2$' results there is a far larger incidence of networks with a higher number of inputs – up to 17 prior values, the additional information absorbed possibly enabling accuracies similar to those obtained when predicting a single interval ahead.

Again, there are several solutions with increased minimum error resulting in reduced maximum error. Further the '23-second' networks exhibit lower mean error and standard deviation when compared to the '1-minute' solutions.

7.2.2 Specific gravity (s.g.) and air/fuel ratio (a/f)

function	*1 interval ahead*	*2 intervals ahead*
linear	0.94 – 1.07	2.27 – 2.52
log	0.84 – 2.56	1.99 – 4.13
tanh	0.84 – 3.55	2.05 – 4.47

Table 7.13 Maximum error ranges (s.g.)

Table 7.13 summarises the maximum error ranges for predicting s.g. by 3 different approaches, whilst table 7.14 shows the error characteristics for possible solutions.

function	*1 interval ahead*				*2 intervals ahead*			
	Struct.	*mean*	*max.*	*s.d.*	*struct.*	*mean*	*max.*	*s.d.*
lsq	—	0.33	2.03	0.34	—	0.86	4.79	0.91
linear	13+K	0.16	0.94	0.14	15+K	0.36	2.27	0.35
log	13+3+K	0.15	0.84	0.13	13+4+K	0.30	1.99	0.29
	13+2+K	0.15	0.92	0.13	12+2+K	0.37	2.19	0.34
tanh	12+3+K	0.14	0.84	0.12	13+3+K	0.33	2.05	0.30
	12+2+K	0.16	0.92	0.13	14+2+K	0.34	2.07	0.31

Table 7.14 Potential solutions (s.g.)

For the s.g. results there is an approximately 200% increase in all parameters when extending the prediction time to two intervals. All networks require some 12 – 15 inputs. There would appear to be an approximate gain of 10% in accuracy by opting for a non-linear solution but the overall accuracy is such that a (more parsimonious) linear solution would suffice. There are no instances of networks with increased mean error providing a reduced maximum error. Benchmarking with the 'least squares' predictor exhibits considerably reduced error levels.

p/p	*function*	*1 interval ahead*	*2 intervals ahead*
none	linear	4.87 – 4.95	4.90 – 5.28
	log	3.66 – 5.92	4.74 – 10.59
	tanh	4.81 – 4.96	3.54 – 12.77
÷10	linear	4.87 – 4.95	4.90 – 5.30
	log	4.84 – 4.90	4.72 – 21.50
	tanh	3.67 – 9.42	4.76 – 7.12
$m = 2$**	linear	4.80 – 4.82	4.68 – 4.70
	log	4.75 – 21.44	4.76 – 14.99
	tanh	4.74 – 526.97	4.64 – 21.34

Table 7.15 Maximum error ranges (a/f)

		1 interval ahead				2 intervals ahead			
p/p	funct.	struct.	mean	max	s.d.	struct.	mean	max	s.d.
lsq	—	—	0.16	5.15	0.26	—	0.40	9.55	0.40
none	linear	3+K	0.11	4.87	0.18	12+K	0.21	4.90	0.29
	log	8+2+K	1.37	3.66	0.97	12+5+k	0.19	4.74	0.31
						14+2+K	0.19	4.82	0.30
	tanh	9+5+K	0.10	4.81	0.17	13+2+K	1.15	3.54	0.79
		8+2+K	0.10	4.86	0.19				
÷ 10	linear	3+K	0.11	4.87	0.18	12+K	0.21	4.90	0.30
	log	8+2+K	0.10	4.85	0.19	13+3+K	0.20	4.72	0.31
						14+2+K	0.21	4.82	0.28
	tanh	10+2+K	1.38	3.67	0.97	13+4+K	0.20	4.76	0.31
						14+2+K	0.21	4.86	0.29
m = 2**	linear	3+K	0.17	4.80	0.21	3+K	0.29	4.68	0.34
	log	8+3+K	0.18	4.75	0.23	14+4+K	0.32	4.64	0.37
		6+2+K	0.16	4.80	0.22	12+2+K	0.31	4.65	0.36
	tanh	10+3+K	0.21	4.74	0.29	13+4+k	0.30	4.64	0.35
		10+2+K	0.20	4.80	0.27	12+2+K	0.28	4.68	0.34

Table 7.16 Potential solutions (a/f)

In the case of the maximum ranges for the a/f results (table 7.15) there is generally little difference between those for the '1-interval ahead' investigation and that for two intervals, with the exception of the '$m = 2$' method which results in exceptionally high upper limit for the tanh activation function. Table 7.16 shows a general increase in the number of network inputs when extending the prediction interval. Further, there are several instances of networks with increased mean error providing reduced maximum error. Again, 'least squares' benchmarking demonstrates the considerably higher error levels of that approach.

7.2.3 Discussion

It is possible to produce acceptable parsimonious linear networks utilising data without pre-processing. There is aliasing present in the synthesised '1-minute' data, indicated by the reduction in mean error and standard deviation within the '23 second' data networks. However, maximum error levels remain similar even when a previously unseen – in terms of training data – major excursion is encountered in the input data. The networks recover from such perturbations (Sections 7.1.2 and A5.3)

The 'division by 100' method of pre-processing tends to track the 'non-processed' networks in terms of accuracy of the results. (However, informal observations during training suggested that training times were shorter for the former method). The '$m = 2$' method of pre-processing results in consistent levels of accuracy of around 4-5% maximum error for all

types of data. It appears to demonstrate resilience in the face of the data excursions described above.

High levels of accuracy were obtained with the s.g. data which was of such a range that did not require any form of pre-processing. The results for a/f ratio were generally of a lower quality than those for the other parameters – greater than 4% maximum error.

There were several instances of networks with substantially increased mean error and standard deviation which resulted in reduced maximum error. This indicates the possibility of over-training and thus the possibility of the validation networks lacking in effectiveness. (Tables 7.9, 7.12 and 7.16.)

In all cases the 'least squares' predictors were out performed by neural network solutions in terms of accuracy, with the former method exhibiting greatly reduced accuracy when predicting two sampling intervals ahead. (Tables 7.9, 7.12, 7.14 and 7.16.)

8.0 HSM Phase 2

Additional data logged at 30 second intervals was supplied, which was closer to the stated mass spectrometer output of approximately 35 second intervals. Hence there was an opportunity to investigate the fuel supply with less aliased data and to compare the results with those of phase 1. However, no details of constituent gases were given in this file; hence it was not possible to deduce which mixing station the gas was being supplied from (See Phase 1, chapter 6).

Furthermore, CV data was not supplied directly but rather in the form of Wobbe numbers. The Wobbe number or index is used as a means of comparing the heat inputs to a burner of different gases at fixed pressure, and is defined as

$$W = \frac{CV}{\sqrt{s.g.}} MJ/kg \qquad \text{——————} (8.1)$$

where CV = calorific value, s.g. = specific gravity

8.1 Analysis of data

The 30 sec data covers an unspecified period during September 2001 and consists of 20,000 points. The parameters of interest to this project are listed in Table 8.1, with graphs of the first three parameters shown in figs. 8.1 – 8.3. (The percentage oxygen content in the exhaust flue forms the basis for the investigation described in chapter 9.)

parameter	range	mean	std. dev.	median
Wobbe number (MJ/kg)	9.47 – 11.6	10.1	0.18	10.09
Specific gravity (s.g.)	0.70 – 0.94	0.78	0.06	0.77
Air/fuel ratio	1.98 –2.76	2.34	0.11	2.35
Oxygen in exhaust flue (%)	0.2 –26.22	3.38	2.54	2.75

Table 8.1 Gas parameter characteristics under investigation

The investigations in this phase of the research proceeded on the following bases:

It had been established by this stage that there was a clear requirement for a minimalist solution in that any implementation would take place on low-specification PLCs salvaged from redundant equipment.

There existed the opportunity to synthesise both 1-minute data for Wobbe numbers and 30-second data for CV using equation 8.1 above, for comparison purposes with Phase 1.

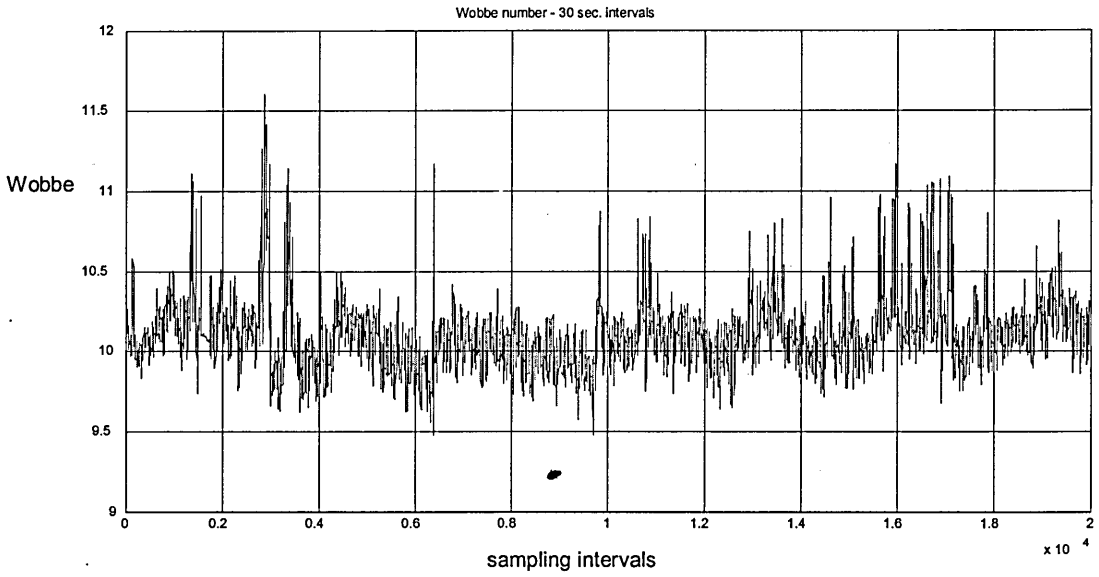


fig. 8.1 Wobbe indices

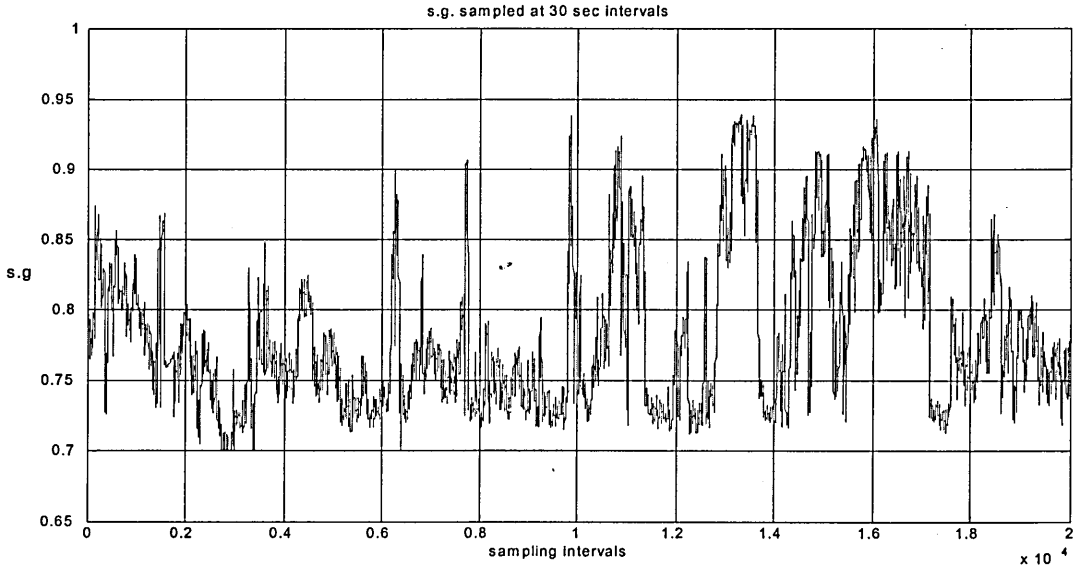


fig. 8.2 specific gravity

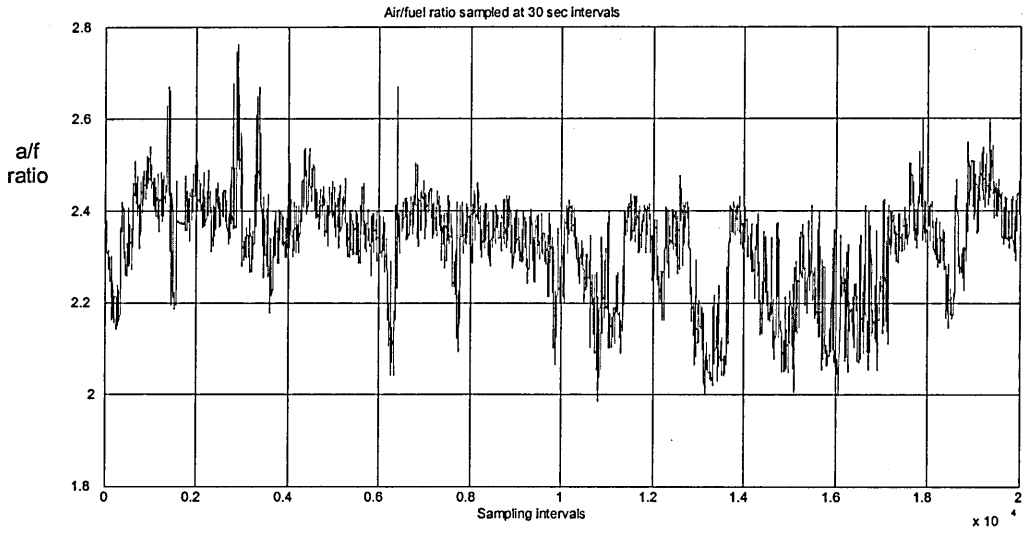


fig. 8.3 air/fuel ratio

Above are figures showing the original 30-second datasets. It was intended to compare the effect of 1-minute and 30-second sampling intervals on accuracy, and investigate the possibility that CV was inherently more difficult to track than Wobbe number.
Thus further data was synthesised as follows:

Wobbe number data sampled at 1-minute intervals, from the 30-second data.

CV datasets sampled at both 30-second and 1-minute details using the supplied s.g. data and equation 8.1 above.

data	*Range*	*mean*	*median*	*s.d.*
Wobbe (30 sec)	9.47 – 11.60	10.1	10.08	0.18
Wobbe (60 sec)*	9.48 – 11.60	10.1	10.09	0.18
Wobbe (Ph1)*	8.51 – 12.74	10.64	10.6	0.47
CV (30 sec)*	8.18 – 10.29	8.93	8.84	0.39
CV (60 sec)*	8.18 – 10.28	8.93	8.84	0.39
CV (Ph.1 60 sec)	7.93 – 11.47	9.75	9.62	0.48

Table 8.2 Comparison of supplied and synthesised(*) data

8.2 Summary of results and discussion

In view of the significant differences both in range and other characteristics between the data sets in Phases 1 and 2, further least-squares-based investigations were conducted for comparison purposes. Those results have been included in the tables below.

8.2.1 Wobbe numbers or indices

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	20.52	17.87	7.95
none	13.08 – 13.14	5.91 – 6.06	3.96 – 4.17
÷100	13.08 – 13.14	5.91 – 6.11	3.96 – 4.17

Table 8.3 Wobbe indices maximum error ranges (1 interval ahead)

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	49.64	35.18	17.8
None	18.88 – 19.00	9.67 – 10.27	7.75 – 8.59
÷100	18.88 – 19.00	9.64 – 10.28	7.75 – 8.60

Table 8.4 Wobbe indices maximum error ranges (2 intervals ahead)

Examination of the results (tables 8.3 and 8.4) for predicting one sampling interval ahead shows that the ranges for the Phase 1 work are very much smaller than those for their Phase 2 counterparts and the accuracy is lower; the maximum error for the Phase 2 data is reduced by some 50% over that in Phase 1. The 30-second data shows a further 30% reduction over the 1-minute data in terms of maximum error. Least accurate of all is the least squares-based method of prediction. Pre-processing through scaling by division by 100 serves little practical purpose.

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	mean	max	s.d.	struct	mean	max	s.d.	struct	mean	max	s.d.
lsq	—	1.53	20.52	1.87	—	0.92	17.87	1.23	—	0.44	7.95	0.58
none	4+K	0.97	13.07	1.11	7+K	0.45	5.91	0.48	12+K	0.22	3.96	0.25
÷ 100	4+K	0.97	13.07	1.11	7+K	0.45	5.91	0.48	12+K	0.22	3.96	0.25

Table 8.5 Wobbe index possible network solutions (1 interval ahead)

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	Mean	max	s.d.	struct	mean	max	s.d.	struct	mean	max	s.d.
lsq	—	3.28	49.64	3.98	—	1.97	35.18	2.59	—	1.03	17.8	1.39
none	5+K	1.80	18.88	1.87	6+K	0.69	9.67	0.70	19+K	0.43	7.75	0.46
÷ 100	5+K	1.80	18.88	1.87	9+K	0.68	9.64	0.69	19+K	0.43	7.75	0.46

Table 8.6 Wobbe index possible network solutions (2 intervals ahead)

The network solutions (tables 8.5 and 8.6) indicate that those from Phase 1 of the HSM work possess fewer cells and therefore fewer inputs, and hence refer to fewer data past values during prediction. In addition they exhibit the highest error parameters. The 30-second training data produces the largest networks, absorbing more of the data 'history' during prediction and hence offer the greatest accuracy.

8.2.2 Calorific value

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	18.13	13.33	7.55
None	11.40 – 11.43	5..69 – 5.77	3.25 – 3.27
÷100	11.40 – 11.44	5..69 – 5.77	3.25 – 3.27
'CV + CO' [m]	11.42 – 11.43	—	—

*least-squares approach [m] multiple inputs of CV and CO

Table 8.7 CV maximum error ranges (1 interval ahead)

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
Lsq*	43.24	26.52	16.28
None	16.78 – 16.99	10.03 – 10.15	7.21 – 7.27
÷100	16.78 – 16.99	10.03 – 10.15	7.23 – 7.32

*least-squares approach

Table 8.8 CV maximum error ranges (2 intervals ahead)

As in the section on Wobbe index prediction above, the least squares-based method offers the least accuracy (tables 8.7 and 8.8). The Phase 1 results appear to indicate that there is little to be gained by the additional CO parameter input. (CO data was not available for the Phase 2 investigations.) The Phase 2 networks yield major improvements : 50% in the case of the 1-minute data over its Phase 1 counterparts, and 58% in the case of the 30-second data over the corresponding 1-minute data, when predicting one sampling interval ahead (tables 8.7 and 8.9). Pre-processing through scaling again appears to serve little practical purpose.

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	mean	max	s.d.	struct	mean	max	s.d.	struct	mean	max	s.d.
Lsq	—	1.31	18.13	1.72	—	0.86	13.33	1.01	—	0.43	7.55	0.51
None	3+K	0.57	11.40	0.71	6+K	0.47	5.70	0.49	3+K	0.22	3.25	0.24
÷ 100	3+K	0.57	11.40	0.71	5+K	0.47	5.70	0.49	3+K	0.22	3.26	0.24
CV + CO	(3+2)+K	0.57	11.42	0.70	—	—	—	—	—	—	—	—

Table 8.9 CV possible network solutions (1 interval ahead)

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	Mean	max	s.d.	struct	mean	max	s.d.	Struct	mean	max	s.d.
Lsq	—	2.85	43.24	3.65	—	1.82	26.52	2.12	—	1.00	16.28	1.19
None	3+K	0.97	16.78	1.11	3+K	0.74	10.03	0.76	3+K	0.44	7.21	0.46
÷ 100	3+K	0.97	16.78	1.11	3+K	0.74	10.02	0.76	3+K	0.44	7.23	0.46

Table 8.10 CV possible network solutions (2 intervals ahead)

When predicting 2 sampling intervals ahead (tables 8.8 and 8.10) there is a similar improvement in performance by the neural networks over the least squares-based approach. For the Phase 1 data the fall in maximum error is 60% and for the Phase 2 data 60% and 55% respectively for the 1-minute and 30-second data. The increase in sampling frequency yields 30% decrease in maximum error with similar performance enhancements for the other error parameters.

8.2.3 Specific gravity

(This data is in a range less than 1.0 and hence no pre-processing was required.)

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	20.52	17.87	7.95
none	13.07 – 13.14	5.91 – 6.06	3.96 – 4.17

*least-squares approach

Table 8.11 Specific gravity maximum error ranges (1 interval ahead)

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq	49.64	35.18	17.8
none	18.88 – 19.00	9.67 – 10.27	7.75 – 8.59

*least-squares approach

Table 8.12 Specific gravity maximum error ranges (2 intervals ahead)

When predicting one interval ahead (table 8.11) the neural network solution in Phase 1 shows an improvement of 36% over the least squares-based method in terms of maximum error. Similarly for Phase 2 the figures are 67% and 50% for the 1-minute and 30-second data respectively. Additionally the higher sampling frequency results in a performance enhancement of 33%.

When predicting 2 sampling intervals ahead (table 8.12) the neural network solution exhibits even greater efficiency reducing maximum error by 62%. For Phase 2 the figures are 73% and 56% for 1-minute and 30-second data with the higher sampling frequency enhancing network performance by 20% - lower than for the case above.

	HSM Phase 1				*HSM Phase 2*							
					1-minute data				*30-second data*			
p.p/ method	*struct*	*mean*	*max*	*s.d.*	*struct*	*mean*	*max*	*s.d.*	*struct*	*mean*	*max*	*s.d.*
lsq	—	1.53	20.52	1.87	—	0.92	17.87	1.23	—	0.44	7.95	0.58
none	4+K	0.97	13.07	1.11	7+K	0.45	5.91	0.48	12+K	0.22	3.96	0.25

Table 8.13 s.g. possible network solutions (1 interval ahead)

	HSM Phase 1				*HSM Phase 2*							
					1-minute data				*30-second data*			
p.p/ method	*struct*	*mean*	*max*	*s.d.*	*struct*	*mean*	*max*	*s.d.*	*struct*	*Mean*	*max*	*s.d.*
lsq	—	3.28	49.64	3.98	—	1.97	35.18	2.59	—	1.03	17.8	1.39
none	5+K	1.80	18.88	1.87	6+K	0.69	9.67	0.70	19+K	0.43	7.75	0.46

**after post-processing

Table 8.14 s.g. possible network solutions (2 intervals ahead)

Comparison of the possible solutions in table 8.14 with those of table 8.13 shows that for the least squares method results the maximum error levels are increased by a factor of two or greater when increasing the prediction interval from 1 to 2. However in the cases of the neural network solutions the respective increases in maximum error levels are 31%, 39%, 49% for each network solution. The Phase 1 networks have the least number of inputs, i.e. make the least reference to historical data, whilst the 30-second trained data has the highest number and the greatest accuracy.

8.2.4 Air/fuel ratio

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	17.98	19.84	13.43
none	13.08 – 13.14	11.36 – 11.67	6.53 – 6.59
÷100	13.08 – 13.14	11.36 – 11.91	6.53 – 6.59

*least-squares approach

Table 8.15 a/f ratio maximum error ranges (1 interval ahead)

p.p./ method	HSM Phase 1	HSM Phase 2	
		1-minute data	30-second data
lsq*	42.49	44.05	30.11
none	18.88 – 19.00	14.05 – 14.56	12.20 – 12.36
÷100	18.88 – 19.00	13.64 – 14.56	12.20 – 12.36

*least-squares approach

Table 8.16 a/f ratio maximum error ranges (2 intervals ahead)

In contrast to all other sets of results obtained so far, the least squares-based solutions offer more accurate solutions for the Phase 1 data than for the Phase 2 1-minute data (tables 8.15 and 8.16). However with reference to tables 8.17 and 8.18, the mean error and standard deviation for the latter solutions are lower in all cases. This is not true of the neural network solutions which continue the trend for more accurate solutions with Phase 2 1-minute data than those for Phase 1.

The least squares-based solutions exhibit more than 200% increase in maximum error when extending the prediction interval to 2 sampling intervals – 236%, 222% and 224% respectively for Phase 1, 1-minute and 30-second Phase 2 data (table 8.18). This contrasts with the corresponding neural network solution figures of 45%, 20%, and 87%.

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	mean	max	s.d.	struct	mean	max	s.d.	Struct	mean	max	s.d.
lsq	—	1.44	17.98	1.72	—	1.17	19.84	1.66	—	0.55	13.43	0.75
none	4+K	0.97	13.07	1.11	8+K	0.59	11.36	0.61	3+K	0.30	6.53	0.38
÷ 100	4+K	0.97	13.07	1.11	7+K	0.59	11.36	0.61	3+K	0.30	6.53	0.38

Table 8.17 a/f ratio possible network solutions (1 interval ahead)

	HSM Phase 1				HSM Phase 2							
					1-minute data				30-second data			
p.p/ method	struct	mean	max	s.d.	struct	mean	Max	s.d.	Struct	mean	max	s.d.
lsq	——	3.15	42.49	3.68	——	2.52	44.05	3.54	——	1.29	30.11	1.81
none	5+K	1.60	18.88	1.67	7+K	0.92	14.05	0.95	3+K	0.62	12.20	0.73
÷ 100	5+K	1.60	18.88	1.67	10+K	0.92	13.64	0.95	3+K	0.62	12.20	0.73

Table 8.18 a/f ratio possible network solutions (2 intervals ahead)

8.2.5 Discussion

In all cases in this section, the least squares-based method produces significantly higher error levels than those of their neural network counterparts; in certain instances the increase is greater than 100%

All methods show significant increases in error levels when the prediction interval is doubled, with typical increases in maximum absolute error of around 100%. This is of particular interest when evaulating the results for the 30-second data trained networks. It should be recalled that the furnace valve trains require some 20 seconds to implement changes in settings. This coupled with the probable presence of large transport lags in the systems may require predictions for two sampling intervals ahead thus nullifying gains in accuracy due to the higher sampling frequency.

It is also instructive to examine the reductions in error levels resulting from doubling the sampling frequency. These are shown in the tables below with the least squares-based results in table 8.19 and those for neural networks trained without pre-processing in table 8.20.

parameter	1-interval ahead			2-intervals ahead		
	mean	max	s.d.	mean	max	s.d.
Wobbe index	-52	-56	-53	-48	-49	-46
CV	-50	-43	-50	-45	-39	-44
s.g.	-52	-56	-53	-48	-49	-46
a/f ratio	-53	-32	-55	-49	32	-49

Table 8.19 Percentage reductions in error levels for 'least squares' predictor after doubling sampling frequency

parameter	1-interval ahead			2-intervals ahead		
	mean	max	s.d.	mean	max	s.d.
Wobbe index	-51	-33	-48	-38	-20	-34
CV	-53	-43	-51	-41	-28	-39
s.g.	-51	-33	-48	-38	-20	-34
a/f ratio	-49	-43	-38	-33	-11	-23

Table 8.20 Percentage reductions in error levels for neural networks (using unpreprocessed data) after doubling sampling frequency

For the neural networks the greatest reductions occur in mean absolute error levels, i.e. the increase in accuracy benefits the majority of the data. Overall there would seem a strong case for further reductions in sampling frequency. However, as discussed in the preceding paragraph there are hardware constraints and a 15 second interval may require predictions of 3 intervals ahead with corresponding loss of accuracy. Further if the resulting networks were incorporated into an on-line closed-loop control system, too high a sampling frequency might result in over frequent activation of the valve trains, with the system never settling to a given set point, and with consequent increases in wear on the valve trains.

Unlike the results for Dawes Lane Coke Ovens (Chapter 7) which indicate that Wobbe index prediction is more efficient than that for CV, the results in this investigation exhibit the opposite trend.

Methods developed using the Phase 2 synthesised 1-minute data are substantially more accurate than those utilising the Phase 1 (sampled at 1-minute intervals) data, often by a factor of 50%. (However, comparison of data in table 6.2 with the data supplied for the Phase 2 work – table 8.1 – shows that the former have wider ranges and higher standard deviations indicating far more 'severe' dynamics.) However there is a single exception with the least squares-based method for air/fuel ratio (Table 8.15).

Scaling through division by 100 offers little improvement in accuracy over unpreprocessed data network training, the differences between the respective results being assumed due to random initialisation of weightings in training.

Overall, the use of pre-trained neural networks to obtain parameters for a linear time series results in some 50% higher accuracy and a much reduced requirement in terms of computational overheads compared to the least squares-based approach.

9.0 Recuperator Corrosion Problem

9.1 Background

Information was received from Corus that they were experiencing a problem with corrosion in the recuperator pipes within a heat exchanger, and a request was made for a brief investigation into developing a predictor along the lines of the previous work, to forecast potentially low oxygen levels in the combustion process which was alleged to be the cause. The suggested indicator was a fall in carbon monoxide level. A spreadsheet was supplied with several sets of data where the oxygen levels in the exhaust flue had fallen to below that which the informant deemed to be desirable – around 1.5%. The author examined these but was not able to identify a specific linkage between fall in CO and consequent fall in O_2.

A decision was made to attempt to track the oxygen levels with a predictor in keeping with previous parameter prediction development, not merely those where the O_2 fell below the acceptable threshold. The data sets supplied were some 30 – 40 points in length. This was inadequate to train, validate and verify a neural network. Examination of the spreadsheet supplied for the HSM Phase 2 work revealed a listing for exhaust flue gas oxygen content, sampled at 30-second intervals and network development proceeded using that data.

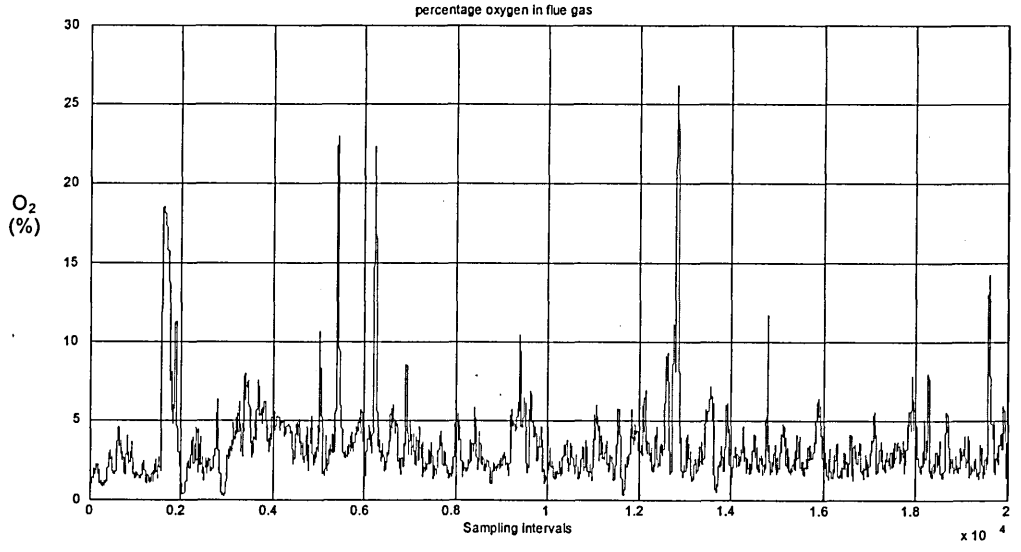


fig. 9.1 Variations in oxygen levels in furnace exhaust gas.

Parameter	*range*	*mean*	*std. dev.*	*median*
Oxygen level in exhast flue (%)	0.2 – 26.2	3.38	2.54	2.75

Table 9.1 Data ranges for exhaust flue oxygen content

9.2 Analysis of data and procedures

The data whilst largely grouped within the range 0 – 6 % shows some significant spikes well in excess of 10%, at intervals 1617 (18.5%), 5416 (22.9%), 6232(22%) and 12821 (26.2%). (It did not prove possible to ascertain the cause of these extreme excursions.) These might hinder the development of a network whose purpose is to detect very low levels of O_2 content.

Hence two strategies evolved : The first developed a linear network along using the techniques in previous chapters. In the second a linear network was trained, validated and verified using low O_2 values only, approximately 10% and lower. Visual inspection of the spikes in figure 9.1 revealed that these last for sometime with gentle slope back to the lower values. (Figure 9.2 below shows the profile for the 26.2% spike at sampling interval 12821.) If it were possible to develop a network which high in accuracy at the critical low values, but less so at the non-critical higher values, the output of the network could be disregarded for values above a certain threshold; the gentle transition from high to low O_2 levels (fig. 9.2) indicated that such a network would have adequate time to resume efficient operation as levels returned to their lower ranges.

purpose	*mins*	*range*	*mean*	*median*	*std. dev.*
training	9001 – 15000	0.28 – 26.22	3.39	2.72	2.58
validation	15001 – 20000	1.26 – 14.23	2.91	2.52	1.45
verification	1- 9000	0.2 – 22.93	3.63	3.0	2.94

Table 9.2 Datasets for first strategy (all data)

Table 9.2 contains details of the datasets chosen for development in strategy 1. Ideally, all sets should exhibit similar characteristics but this did not prove possible with the available data. However, the training and validation sets cover almost the entire data range, whilst their lowest value is similar to that of the verification set. Table 9.3 details the datasets for strategy 2.

purpose	*mins*	*range*	*mean*	*median*	*std. dev.*
Training	9449 – 12798	0.28 – 11.12	3.25	2.93	1.73
Validation	12880 – 19580	0.48 – 11.69	2.80	2.46	1.24
Verification	1900 – 5000	0.2 – 8.0	3.58	3.58	1.6

Table 9.3 Datasets for second strategy (low O_2 only)

No pre-processing of data was undertaken in either case, and only values one sampling interval ahead were predicted. A least squares-based predictor simulation was performed to provide benchmarking data.

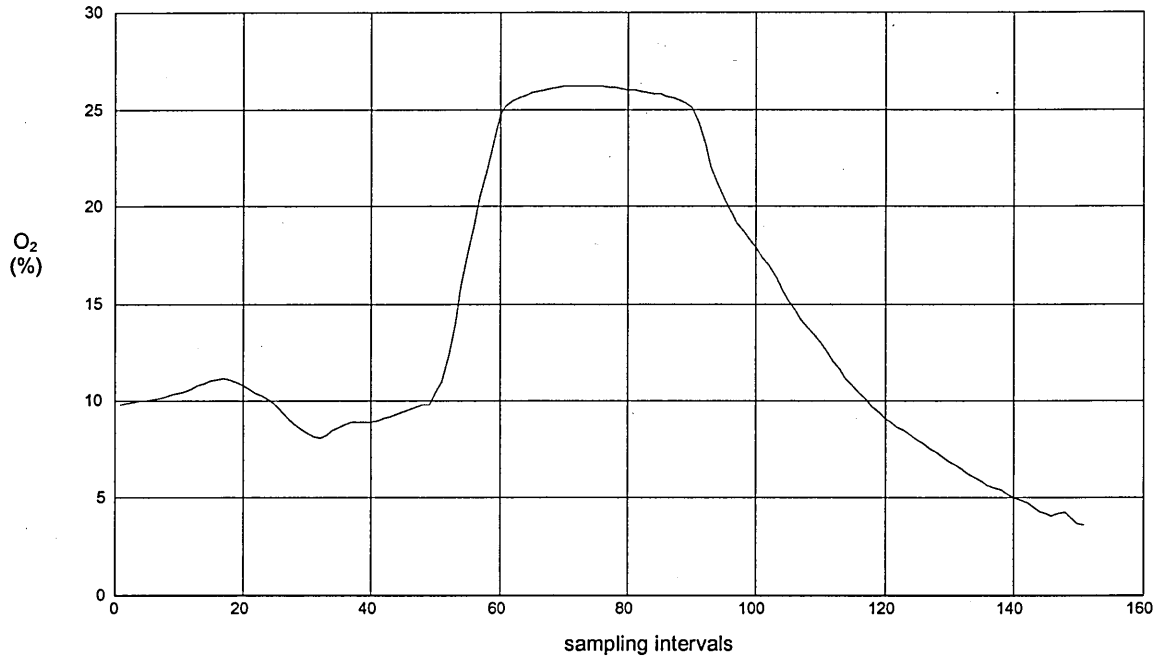


fig. 9.2 close-up of spike at sampling interval 12821 mins

9.3 Results and discussion

Tables 9.4 and 9.5 record typical experimental results, with error parameters summarised in table 9.6. Tables 9.7 suggests 4-input network solutions for both approaches, with maximum absolute error magnitudes 15.8% and 15.4% respectively. Whilst these figures are similar, the mean error levels and standard deviation are higher for strategy 2, limited to lower level O_2 values. (It should be noted that because of this limitation there were fewer data points available.)

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0122	0.1656	0.0136	0
*4	0.0125	0.1580	0.0134	0
5	0.0127	0.1642	0.0137	0
6	0.0128	0.1651	0.0138	0
7	0.0128	0.1676	0.0139	0
8	0.0128	0.1683	0.0140	0
9	0.0128	0.1684	0.0140	0
10	0.0128	0.1684	0.0140	1
11	0.0129	0.1700	0.0141	0
12	0.0129	0.1694	0.0142	1
13	0.0129	0.1704	0.0142	1
14	0.0129	0.1704	0.0142	2
15	0.0129	0.1695	0.0142	1
16	0.0129	0.1683	0.0142	1
17	0.0129	0.1681	0.0142	1
18	0.0129	0.1689	0.0142	1
19	0.0129	0.1692	0.0142	2
20	0.0129	0.1691	0.0142	1

Table 9.4 Strategy 1 (all data)

inputs	*Mean*	*max*	*s.d.*	*<0.01*
3	0.0129	0.1571	0.0160	0
*4	0.0129	0.1544	0.0159	0
5	0.0129	0.1567	0.0159	0
6	0.0130	0.1666	0.0161	0
7	0.0134	0.1662	0.0164	0
8	0.0133	0.1670	0.0164	0
9	0.0135	0.1813	0.0166	1
10	0.0135	0.1815	0.0166	2
11	0.0136	0.1886	0.0169	0
12	0.0137	0.1917	0.0170	1
13	0.0137	0.1906	0.0170	2
14	0.0137	0.1894	0.0169	1
15	0.0137	0.1878	0.0170	2
16	0.0137	0.1834	0.0169	1
17	0.0137	0.1840	0.0170	1
18	0.0137	0.1806	0.0169	1
19	0.0137	0.1812	0.0169	0
20	0.0137	0.1783	0.0169	2

Table 9.5 Strategy 2 (low O_2 only)

Tables 9.4 and 9.5 Typical results

	error (%)		
method	*mean*	*max*	*std. dev*
lsq*	4.54	71.04	4.82
all data	1.22 - 1.29	15.8 - 17.04	1.34-1.42
low	1.29 – 1.37	15.44 - 19.17	1.59 – 1.70

*least squares-based simulation

Table 9.6 Error ranges for tables 9.4 and 9.5

		error (%)		
method	*Structure*	*mean*	*max*	*std. dev*
lsq*	—	4.5	71	4.8
all data	4+K	1.3	15.8	1.3
low	4+K	1.3	15.4	1.6

*least squares-based simulation

Table 9.7 Comparison of potential solutions

Examination of the neural network solutions in table 9.7 reveals that the network derived from low O_2 data only, provides little in the way of enhanced accuracy, merely a marginal reduction in maximum error at the expense of increased standard deviation. Both neural networks exhibit improvements in excess of 70% in terms of mean and maximum error when compared to the least squares-based simulation, thus emphasising the superiority of neural network approach within this context.

10.Review, Conclusion, and Further Work

10.1 Review

Both linear and non-linear minimalist neural network solutions were investigated and their performance evaluated against equivalent least squares-based predictors.

Data from a variety of sources was investigated:

(a) *Single gas mix COG (Appendix 1)*

Two sets of data supplied from two different sources.

(i) A user – Central Power Station (CPS, chapter 5) – predicting CV only, and sampled at 1-minute intervals from the main archiving systems.

(ii) A producer – Dawes Lane Coke Ovens (chapter 7) – predicting CV, s.g., and air/fuel ratio. In this case data was supplied directly from the mass spectrometer output at 23-second intervals. CV data was synthesised at 1-minute intervals for comparison with the CPS results (chapters 5 and 7).

(b) Complex gas mix MEG (Appendix 1)

Two sets of data from the Heavy Section Mill (HSM), using Mixed Enhanced Gas supplied via a complex gas grid (chapter 6).

(i) The initial set (chapter 6) emanated from the 1-minute sampled main archival system covering CV, s.g., and air/fuel ratio. Wobbe index data was calculated from CV data for comparison with the results obtained in (b) below. The addition of secondary percentage gas content parameters (oxygen and hydrogen) as a means of indicating gas supplier, did not improve network accuracy.

(ii) A second series of data (chapter 8) was provided directly from the mass spectrometer output, sampled at 30-second intervals, and consisting of Wobbe indices, s.g., and air/fuel ratio. For each of these parameters, '1-minute' data was synthesised to enable comparison with the work in (a) above (chapter 8). Similarly, CV data was calculated from Wobbe index information and again '1-minute' data sythesised.

(c) Flue gas percentage oxygen content

A secondary investigation (chapter 9) was conducted into the feasibility of applying the neural network methodology to predicting low oxygen content in the exhaust flues of the HSM, which was understood to be a contributing factor in recuperator corrosion. Data sampled at 1-minute intervals was obtained from the HSM archival system.

Where minimalist non-linear networks were investigated (chapters 5 and 7), their performance was found to be equivalent, or inferior, to the equivalent linear networks.

It was established that pre-processing of data via scaling was unnecessary (chapters 5-8). In the case of division by m, the mean of the past n values, it was found that the optimum accuracy was obtained when $n = 2$ (referred to as '$m = 2$' in this work). However that method consistently resulted in lower accuracy especially when applied to the rapidly fluctuating data of the HSM with information loss due to the inherent smoothing effect of the method (chapters 5-7). Scaling through division by a constant (100) – '÷100' – had only minimal effect on accuracy when compared to the output of networks trained with unprocessed data (chapters 5-8).

Least squares-based predictor performance when predicting CV 1 sampling interval ahead on relatively slowly changing data such as COG was comparable to that of linear network solutions (chapter 7). However error levels increased substantially when predicting two intervals ahead, or when aliasing was introduced through increasing the sampling interval to 1 minute (tables 7.8 and 7.11). In contrast, examination of the s.g. and air/fuel results for COG indicate substantially higher levels for all error parameters in respect of the 'least squares' approach (tables 7.14 and 7.16).

The results from the more rapidly fluctuating data levels of the MEG parameters (chapter 6) further emphasised the superior accuracy of the neural network approach with reductions in error levels of 50% or greater over the equivalent 'least squares' predictors (chapter 8). Particularly noteworthy was the ca. 75% reduction in maximum error (table 8.14) for s.g. (two sampling intervals ahead, '1-minute' data), attained by the neural network predictor. (That data contained aliasing deliberately introduced by synthesising '1-minute data' from that originally sampled at 30-second intervals.) Addition of secondary network inputs of

percentage gas content (oxygen and hydrogen) as a means of indica.ting gas supplier, did not improve accuracy over that attained by simple CV-tracking networks

Finally, the results (table 9.7) obtained when predicting oxygen content in flue gas (sampled at 1-minute intervals) using a linear neural network show very substantial reductions in mean error (66%), maximum error (ca. 80%) and standard deviation (ca. 75%) over the levels attained by the least squares approach.

10.2 Conclusion

Minimalist linear neural networks capable of predicting values up to two sampling intervals ahead were successfully developed.

Where an additional non-linear network investigation was performed, the corresponding linear solution was found to be equivalent or superior, in performance.

When measured against a benchmark least squares-based predictor, the neural network approach resulted in substantially higher accuracy when predicting two intervals ahead, or when confronted with rapidly changing and/or aliased data .

It is possible to develop networks of high accuracy without recourse to pre-processing of data during training.

The networks developed in this work are eminently suited to implementation on low specification, possibly nominally redundant, hardware (e.g. a PLC) prolonging the life of equipment and reducing the need for new investment, with consequent economic advantages. Additionally, there are environmental benefits in the re-use, rather than the immediate disposal, of nominally redundant equipment.

11. Further Work

A "shadow" implementation of the results from this research is proposed, running in parallel to the existing control methods with data being logged over a period of several months, in order to further validate the method. Predicted values would be compared with actual values, and calculations would be performed with respect to the corresponding controller settings (proposed and actual) in order to evaluate possible enhancements and efficiency savings.

However, several of the departments at Corus' Scunthorpe site have now ceased operation including the HSM, and alternative arrangements would have to be sought.

Investigation into the economic and environmental benefits of such minimalist solutions should be extended to other (non-steelworks related) processes, e.g. prediction of flow rates, gas content, temperature, especially where aliasing appears to be present or data is incomplete.

References

Bocc99	Boccaletti C, Cerri G, Khatri DS, Syedan B. *An application of neural network in combustion process evaluations.* International Conference on Enhancement and Promotion of Computational Methods in Engineering and Science 7, Macao, August 1999
Chawdry89	Chawdry PK, Hogg BW. *Identification of Boiler Models.* IEE Proceedings Pt. D : Control Theory and Applications, vol 136 pt5, 1989
Cui92	Cui X, Kang GS. *Application of Neural Networks to Temperature Control in Thermal Power Plants.* Engineering Applications of Artificial Intelligence, vol. 5, no. 6, 1992
Durk94	Durkin J.*Expert Systems: design and development.* Prentice Hall, 1994, ISBN 0023309709
Eastop90	Eastop T D, Croft D R. *Energy Efficiency for Engineers and Technologists.* Longman, 1990, Reprinted 1996, ISBN 0582031842
Elan95	Elanayar S, Shin YC. *Robust Tool Wear Estimation with Radial Basis Function Neural Networks.* Journal of Dynamic Systems, Measurement, and Control, vol. 117, December 1995
Farnell	Farnell Ltd (http://www.farnell.com)
Faus94	Fausett L. *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications.)* Prentice Hall, 1994
Giar98	Giarratano Joseph C.; Riley, Gary. *Expert systems: principles and programming.*, PWS Publishing, 1998, ISBN 0534950531 (Includes CLIPS software)
Gib96	Gibson GL, Brandon A. *Using Artificial Intelligence to Improve Power Plant Performance.* 39th Power Instrumentation Symposium. Baltimore, June 1996
Hagan95	Martin T. Hagan, Howard B. Demuth, Mark Beale. *Neural Network Design.* PWS, 1995
Hayk99	"Neural Networks (A Comprehensive Foundation)", Simon Haykin, 2ed., 1999
Khalil92	Khalil M, Omatu S. *A Neural Network Controller for a Temperature Control System.* IEE Control Systems Magazine. June 1992
KISS86	Kissell TE. *Understanding and Using Programmable Controllers.* Prentice Hall International, 1986
Krost96	Krost G, Spanel U, Müller C. *Self-acting Expert Systems in Electric Energy System Operation.* IEEE International Conference on Intelligent Systems Applications to Power Systems. Orlando, USA. February 1996
Krost98	Krost G, Rumple D, Spanel U, Müller C, Ju K. *Autonomous Expert System Operation in Electrical Energy Systems.* International Journal of Engineering Intelligent Systems, vol. 6 pt 1, March 1998
Ladd95	Scott Robert Ladd. *Genetic Algorithms in C++.* M & T Books, 1995. ISBN 1558514597
Lown97	Lown M, Swidenbank E, Hogg BW. *Adaptive Fuzzy Control of a Pilot Turbo-Generator Plant.* Proceedings of the 1997 IEE Colloquim on Intelligent Control, London 1997
Lu94	Lu S, Hogg BW, Ni WD. *Object-oriented Power Plant and Control System Modelling.* IEEE International Conference on Power System Technology, Beijing 1994
MANO88	Mano MM. *COMPUTER ENGINEERING : Hardware Design.* Prentice Hall International, 1988
Maplin	Maplin Ltd (http://www.maplin.com)

Nar90 — Narenda KS, Parthasathy K. *Identification and Control of Dynamical Systems using Neural Networks.* IEE Transactions on Neural Networks, vol. 1, no. 1, March 1990

Ordys94 — Ordys AW, Pike AW, Johnson MA, Katebi RM, Grimble MJ. *Modelling and Simulation of Power Generation Plants.* Springer-Verlag, London, 1994

Rao95 — Rao, Valluru B. *C++ neural networks and fuzzy logic.* M&T Books, 2ed., 1995

Rein91 — Reinschmidt KF
"Neural networks: next step for simulation and control", Power Engineering, Nov 1991

Rein94 — Reinschmidt KF. *Neural Networks for Plant Simulation and Control.* 56th Annual American Power Conference, vol. 56, pt.1. Chicago, April 1994

Rein95a — Reinschmidt KF *Artificial Neural Networks for Short Term Load Forecasting.* 57th Annual American Power Conference, Chicago, April 1995

Rein95b — Reinschmidt KF *Artificial Neural Networks in Short Term Load Forecasting.* IEEE Conference on Control Applications, Albany, Sept 1995

Rösch92 — Roescheisen M, Reimar H, Tresp V. *Neural Control for Rolling Mills: Incorporating Domain Theories to Overcome Data Deficiency*". Advances in Neural Information Processing, vol.14, 1992

RS Components — RS Components Ltd. (http://rswww.com)

Sanz95 — Sanz R, Rajoy JA, Bernadox J, Farina J. *Knowledge Based Systems for Daily Generation Scheduling of a Small Hydropower Plant.* 10th International Conference on Applications of Artificial Intelligence in Engineering, Udine, Italy, July 1995

Shalk97 — Schalkoff , Robert J. *Artificial Neural Networks.* McGraw-Hill, 1997

SNNS98 — Manual, Stuttgart Neural Network Simulator. (http://www-ra.informatik.uni-tuebingen.de/SNNS/)

A.1 Gas Properties

		COG	BFG	BOS	Natural Gas
Carbon monoxide	CO	6	25	58.5 – 80.2	—
Carbon dioxide	CO_2	2.41	25	9.00 – 16.77	0.1
Hydrogen	H_2	51.47	3.5	0.57 – 1.22	—
Nitrogen	N_2	7.17	46.5	9.61 – 22.98	2.6
Argon	A	0.05	—	0.22 – 0.27	
Methane	CH_4	29.94	—	—	92.6
Ethylene	C_2H_4	1.67	—	—	
Ethane	C_2H_6	0.82	—	—	3.6
Oxygen	O_2	0.05	—	—	
Hydrogen sulphide	H_2S	0.1	0.00092	—	
Benzine	C_6H_6	trace	—	—	
Toluene	C_7H_8	trace	—	—	
Propylene	C_3H_6	0.08	—	—	
Acetylene	C_2H_2	0.05	—	—	
Propane	C_3H_8	0.09	—	—	0.8
Calorific values (MJ/Nm^3)		19	3.5	7.5 – 10	38.6
Specific gravity		0.38	1.073	1.0 – 1.05	
Flammability (% in air)	lower	4.5	32.8 – 55.8 (usually 38.9)	16.2	
	Higher	37.2	69.5 – 70.1 (usually 70.1)	70.5	

Table A1.1 Typical percentage constituents and calorific values

COG has a lower CO_2 content which improves furnace efficiency, high H_2 which improves flame speed and stability and lower nitrogen than BFG, improving furnace efficiency. However COG produces the pollutant sulphur dioxide (SO_2).

However the amount of COG produced at the Scunthorpe site is insufficient for all requirements and has at at times of high demand to be supplemented by Synthetic Coke Oven Gas (SCOG) which is produced by mixing BFG and Natural Gas. No details were supplied in respect of the mixing ratio and percentage constituents, but some details for Natural Gas alone were obtained from Eastop [Eastop90, p76, p353] and are listed in the rightmost column. It is assumed that the resulting mix would probably be high in methane content, and have a far higher cv than BFG alone.

It is unclear how a large proportion of SCOG rather than COG would affect the gas in the supply main to the HSM, nor if it were actually present during the period covered by the supplied data.

		Central	*Anchor*
Carbon monoxide	CO	14.74	60.78
Carbon dioxide	CO_2	13.41	12.63
Hydrogen	H_2	28.88	9.17
Nitrogen	N_2	29.10	13.23
Argon	A	0.74	0.34
Methane	CH_4	11.5	3.108
Ethylene	C_2H_4	0.93	0.258
Ethane	C_2H_6	0.42	0.084
Oxygen	O_2	0.05	0.33
Hydrogen sulphide	H_2S	0.16	0.023
Benzene	C_6H_6	—	—
Toluene	C_7H_8	—	—
Propylene	C_3H_6	0.035	0.015
Acetylene	C_2H_2	0.016	0.007
Propane	C_3H_8	0.83	0.022
Calorific values (MJ/Nm^3)		10.06	10.14
Specific gravity		0.738	0.945

Table A1.2 Comparison of the Mixed Enhanced Gas (MEG) percentage constituents as supplied by the Central and Anchor mixing stations

The most significant constituent in the Anchor gas mix is CO at 61%, four times that of Central. Whereas Central's mix is rich in H_2 and N_2. (With methane levels at four times those of Anchor, it is assumed that Central receives a higher proportion of COG – see Table A1.1 above) The respective levels of CO, H_2 and N_2 should serve as indicators as to which mixing station is the predominant supplier, if any, at a given moment.

Appendix A.2 - Network Training

(The Method of Backpropagation of Errors, and the Delta Rule)

The *back-propagation of errors*, and the *generalised delta rule* (GDR) training algorithms appear to have been developed independently by several researchers, e.g. Rummelhart, McClelland, and others [Hayk99, Faus94], and is in part derived from the method of least mean squares. It is a steepest descent method and seeks to apply a correction Δw_{ji} to the 'synaptic' weight linking input y_i to the j^{th} cell in the network, which is proportional to $\partial E/\partial w_{ji}$ the sensitivity of the output error to incremental changes in Δw_{ji} at the p^{th} training iteration, or

$$\Delta w_{ji}(p) = -\eta \, \partial E / \partial w_{ji}(p) \qquad \text{——— } (A2.1)$$

This is known as the *Delta Rule.* The minus sign indicates a descent in error space and η is a positive constant termed the gain or *learning rate.*

It is characterised by two passes through the network : forward where the network outputs are calculated and compared with actual *target* values and the differences or errors, computed. These errors are then *backpropagated* in a backward pass through the network layer by layer and adjustments to the cell input weightings made in accordance with the above equation.

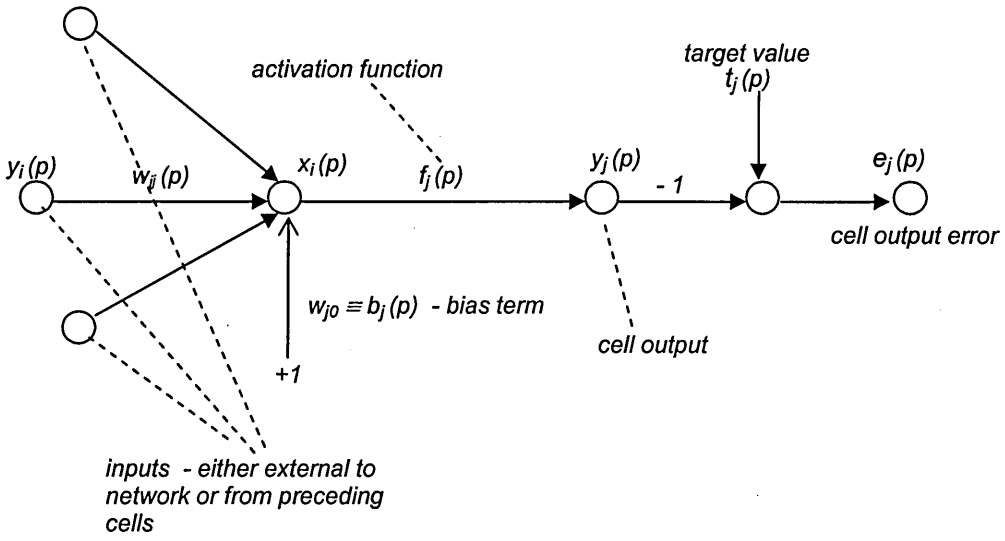


fig. A2.1 Signal flow diagram for cell in output layer

Whilst target (desired) values are available for cells in the network output layer and the error computed easily, no such values are directly available for intermediate (hidden) layers although the weightings within these layers contribute to the effectiveness of the network. A different method utilising the back-propagation of errors is employed, and accordingly the two cases are treated separately in this description.

A2.1.1 Cells in the output layer

Consider the diagram (fig. A2.1 above) showing the j^{th} cell in an output layer at the p^{th} iteration of training. Each input-output pattern is presented to the network and the error *e(p)* computed.

$$e_j(p) = t_j(\mathrm{p}) - y_j(p) \quad \text{——— } (A2.2)$$

The instantaneous error energy for the j^{th} cell is defined as $\frac{1}{2}e_j^{\,2}(p)$ and for all n cells in the output layer

$$E(p) = \frac{1}{2}e_1^{\,2}(p) + \frac{1}{2}e_2^{\,2}(p) + \ldots + \frac{1}{2}e_j^{\,2}(p) + \ldots + \frac{1}{2}e_n^{\,2}(p) \quad \text{——— } (A2.3)$$

$$E(p) = \frac{1}{2}\sum_{j=1}^{n} e_j^{\,2}(p) \quad \text{——— } (A2.3a)$$

For a training set containing N input-output patterns, then over one *epoch* (one pass through the entire training data) the *mean squared* error is given by

$$E_{av} = \frac{1}{N}\sum_{p=1}^{N} E(p) \quad \text{——— } (A2.4)$$

This serves as a measure of performance and is the cost function to be minimised during training.

For the j^{th} cell the sum of the weighted m input signals is given by

$$x_j(p) = \sum_{i=0}^{m} w_{ji}(p) y_i(p) \quad \text{——— } (A2.5)$$

where y_i is an input signal which may be a direct input to the network or the output of a cell in the previous layer. The bias input b_j is expressed as a constant input $y_0 = 1$ with weighting w_{j0}.

The output from that cell's activation function $f_j(.)$ is

$$y_j(p) = f_j\left(x_j(p)\right) \quad \text{——— } (A2.6)$$

Now from the delta rule above (*A2.1*) the correction to be applied to weight w_{ji} at the p^{th} training iteration is

$$\Delta w_{ji}(p) = -\eta\, \partial E(p) / \partial w_{ji}(p) \quad \text{——— } (A2.1)$$

The derivative term may be expanded via the calculus chain rule to incorporate responses to incremental changes in

- network error $E(p)$ with respect to changes in cell output error $e_j(p)$
- cell output error $e_j(p)$ with respect to changes in cell output $y_j(p)$
- cell output $y_j(p)$ with respect to changes in total input signal level $x_j(p)$
- total input signal level $x_j(p)$ with respect to changes in weighting $w_{ji}(p)$

Hence

$$\frac{\partial E(p)}{\partial w_{ji}(p)} = \frac{\partial E(p)}{\partial e_j(p)} \frac{\partial e_j(p)}{\partial y_j(p)} \frac{\partial y_j(p)}{\partial x_j(p)} \frac{\partial x_j(p)}{\partial w_{ji}(p)} \quad \text{——— } (A2.7)$$

And from eqn. (*A2.3*)

$$\frac{\partial E(p)}{\partial e_j(p)} = e_j(p) \quad \text{——— } (A2.8)$$

From (*A2.2*)

$$\frac{\partial e_j(p)}{\partial y_j(p)} = -1 \quad \text{——— } (A2.9)$$

From (*A2.6*)

$$\frac{\partial y_j(p)}{\partial x_j(p)} = f'\left(x_j(p)\right) \quad \text{——— } (A2.10)$$

where $f'(x_j(p))$ is the first derivative of the cell's activation function. For certain activation functions there exists a simplified relationship with the derivative which greatly facilitates computation with minimal overheads. (Section *A3.1*)

From (*A2.6*)

$$\frac{\partial x_j(p)}{\partial w_{ji}(p)} = y_i(p) \quad \text{——— (A2.11)}$$

Substituting (*A2.18*) to (*A2.11*) in (*A2.7*)

$$\frac{\partial E(p)}{\partial w_{ji}(p)} = -1.e_j(p).f_j\big(x(p)\big).y_i(p) \quad \text{——— (A2.12)}$$

At this point a local gradient $\delta_j(p)$ due to the effect of the j^{th} cell within the error field for the network is defined as the sensitivity of the total network output error to incremental changes in the sum of input signals to cell j^{th}, i.e.

$$\delta_j(p) = -\frac{\partial E(p)}{\partial x_j(p)} \quad \text{——— (A2.13)}$$

which expands by chain rule to

$$\delta_j(p) = -\frac{\partial E(p)}{\partial e_j(p)}\frac{\partial e_j(p)}{\partial y_j(p)}\frac{\partial y_j(p)}{\partial x_j(p)} \quad \text{——— (A2.14)}$$

and using the results gained for those partial derivatives, above

$$\delta_j(p) = e_j(p).f_j'\big(x_j(p)\big) \quad \text{——— (A2.15)}$$

Substituting this result into (*A2.12*) gives

$$\frac{\partial E(p)}{\partial w_{ji}(p)} = -\delta_j(p).y_i(p) \quad \text{——— (A2.16)}$$

Thus for a cell in the network output layer the Delta Rule (*A2.1* above) may be expressed as

$$\Delta w_{ji}(p) = -\eta.\delta_j(p).y_i(p) \quad \text{——— (A2.17)}$$

A2.1.2 Cells in a Hidden Layer

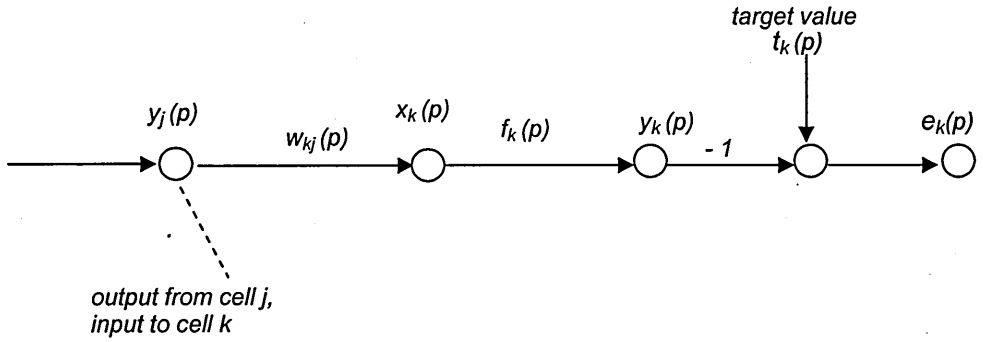


fig. A2.2 Simplified signal flow diagram for cell k in output layer fed from cell j in hidden layer

Figure A2.2 depicts a situation in which cell j above lies within a hidden a hidden layer feeding the k^{th} cell of an output layer, and connected to that cell via a weight w_{kj}.

Equation (*A2.3a*) above may be re-written for the k^{th} cell as

$$E(p) = \frac{1}{2}\sum_k e_k^{\,2}(p) \quad \text{——— (A2.18)}$$

and similarly, re-writing and combining (*A2.2*) and (*A2.6*)

$$e_k(p) = t_k(p) - y_k(p) = t_k(p) - f_k\big(x_k(p)\big) \quad \text{——— (A2.19)}$$

where the input to the activation function f_k is defined per (*A2.5*) above as

$$x_k(p) = \sum_{j=0}^{m} w_{kj}(p) y_j(p) \quad \text{——— (A2.19a)}$$

The local gradient $\delta_j(p)$ is re-defined for the j^{th} cell in the hidden layer with a different chain rule expansion without reference to cell output error $e_j(p)$.

$$\delta_j(p) = -\frac{\partial E(p)}{\partial x_j(p)} = -\frac{\partial E(p)}{\partial y_j(p)} \frac{\partial y_j(p)}{\partial x_j(p)} \quad \text{——— (A2.20)}$$

which from (*A2.10*) becomes

$$\delta_j(p) = -\frac{\partial E(p)}{\partial y_j(p)} f'\big(x_j(p)\big) \quad \text{——— (A2.21)}$$

A link between the hidden j^{th}cell and the k^{th} output cell, for which the error is known, is established by differentiating (*A2.18*) with respect to the j^{th} cell's output y_j

$$\frac{\partial E(p)}{\partial y_j(p)} = \sum_k e_k(p)\frac{\partial e_k(p)}{\partial y_j(p)} = \sum_k e_k(p)\frac{\partial e_k(p)}{\partial x_k(p)}\frac{\partial x_k(p)}{\partial y_j(p)} \quad \text{——— (A2.22)}$$

again by chain rule and noting the introduction of the additional term x_k – the sum of the k^{th} cell's input signals, i.e. the total input to its activation function.

From (*A2.19*)

$$\frac{\partial e_k(p)}{\partial x_k(p)} = -f_k'(x(p)) \quad \text{——— (A2.23)}$$

And from (*A2.19a*)

$$\frac{\partial x_k(p)}{\partial y_j(p)} = w_{kj}(p) \quad \text{——— (A2.24)}$$

Hence

$$\frac{\partial E(p)}{\partial y_j(p)} = \sum_k e_k(p)\left(-f_k'(x_k(p))\right)w_{kj}(p) = -\sum_k e_k(p)f_k'(x_k(p))w_{kj}(p) \quad \text{— (A2.25)}$$

Now from (*A2.15*) the local gradient for the k^{th} output cell

$$\delta_k(p) = e_k(p).f_k'(x_k(p)) \quad \text{——— (A2.26)}$$

Substituting this result in (*A2.25*)

$$\frac{\partial E(p)}{\partial y_j(p)} = -\sum_k \delta_k(p)w_{kj}(p) \quad \text{——— (A2.26)}$$

Thus from (*A2.21*) the local gradient for the j^{th} cell in a hidden layer is given by

$$\delta_j(p) = f'(x_j(p))\sum_k \delta_k(p)w_{kj}(p) \quad \text{——— (A2.27)}$$

i.e. the product of the derivative of the activation function and the summed products of the local gradients of the cells in the next layer, fed by that cell, and the respective connecting weights.

A2.1.3 In general

For cells in the both the output and hidden layers, Haykin summarises the above as

$$\begin{pmatrix} weight \\ correction \\ \Delta w_{ji} \end{pmatrix} = \begin{pmatrix} learning \\ rate \\ \eta \end{pmatrix} . \begin{pmatrix} local \\ gradient \\ \delta_j(p) \end{pmatrix} . \begin{pmatrix} neuron \\ input \\ y_i(p) \end{pmatrix}$$

[Hayk99, p166]

Or,

$$\Delta w_{ji} = \eta \delta_j(p) y_i(p) \quad \text{——} \quad (A2.28)$$

For the j^{th} cell in a layer at the p^{th} training iteration. For cells the output layer, the local gradient is given by

$$\delta_j(p) = e_j(p).f_j'\left(x_j(p)\right) \quad \text{——} \quad (A2.29)$$

which contains actual information about the network output error. For cells in the hidden layer(s) the local gradient is given by

$$\delta_j(p) = f'\left(x_j(p)\right)\sum_k \delta_k(p) w_{kj}(p) \quad \text{——} \quad (A2.30)$$

which does not directly contain network output error information. Instead the error information is propagated back through the network by local gradients in the next layer.

A2.2 The generalised delta rule learning rate, and momentum

Schalkoff remarks that equation *(A2.28)* is based on a first-order gradient descent method and therefore may

> "... find a local minimum in" ...error space *E* which " .... may correspond to suboptimal solutions with respect to the global minima."

Varying the learning rate η may diminish the possibility of this but its should be noted that for

small η
- there are small changes to weights, but a smoother trajectory in weight space. However there is the risk that the function will become 'trapped' within a local minimum.
- training is slow.

large η
- training is fast and 'steps' between changes to weightings are sufficiently large to result in a 'trapped' function to emerge from a local minimum.
- the resulting large changes in weights may make network unstable – oscillation may occur may occur around the optimum solution – and the function may 'jump over' and miss the actual global minimum.

Schalkoff proposes a variable learning rate solution, in which training is commenced with a large learning rate $\eta(0)$ and is reduced after each epoch according to the formula

$$\eta(p) = \frac{\eta(p)}{p} \quad \text{——— (A2.31)}$$

The Delta Rule above can be refined to overcome this to a large extent by including a (positive) *momentum* constant α

$$\Delta w_{ji}(p) = \eta \delta_j(p) y_i(p) + \alpha \Delta w_{ji}(p-1) \quad \text{——— (A2.32)}$$

i.e. the current weight adjustment is now also dependent on that previously obtained. Equation (*A2.28*) forms the *generalised delta rule.*

Two significant aspects of momentum with respect to $\partial E / \partial w_{ji}$, the error sensitivity – see equations (*A2.1*) and (*A2.7*) – are

– when $\partial E / \partial w_{ji}$ has the same algebraic sign on consecutive iterations, w_{ji} is modified by a large amount. Thus, momentum tends to accelerate descent in steady downhill directions (ie, giving momentum to the correction).

– when $\partial E / \partial w_{ji}$ has alternating algebraic signs on consecutive iterations, w_{ji} becomes smaller and so the weight adjustment is small. Thus, momentum has a stabilising effect on learning.

A2.3 Potential deficiencies in backpropagation

For a cell in the output layer, from (*A2.17*) substituting from (*A2.15*)

$$\Delta w_{ji}(p) = -\eta . e_j(p) . f_j'\left(x_j(p)\right) y_i(p) \quad \text{——— (A2.17)}$$

If the input $y_i(p)$ to cell j is zero, or near zero, then the weight correction will be zero or near zero.

Correction is also dependent on the derivative of the activation function. It will be minimised for a non-linear activation function when the cell is operating in its saturation zones, i.e. at the extremes where the curve flattens and the derivative becomes small.

In the case of hidden cells where

$$\delta_j(p) = f'\left(x_j(p)\right) \sum_k \delta_k(p) w_{kj}(p) \quad \text{——— (A2.27)}$$

there is also a dependency on the activation function derivative.

Schalkoff comments that it illustrates

"... the necessarily serial nature of the GDR ... algorithm. First, note the possibility of premature saturation in hidden-layer units. However, there is an even more significant possible shortcoming due to the recursive formulation. Since... " (eqn z computes δ_h) "... additional concerns are as follows:

1. The effect of small δ in a previous layer (perhaps due to premature saturation) could be exaggerated in the cascaded combination of two saturated units (*cells*). If unit *h* is saturated the weighting" ... (*due to* $f_h'(x_h (p))$) ... "further reduces the correction.
2. This process may continue backward, from the output layer, until weight corrections are in consequential. This partially explains why networks with large numbers of hidden layers train poorly using the GDR strategy." [Shalk97, p158]

Appendices A3-6 – Tables of typical experimental results

Notation used when reporting neural network results

The following abbreviations and conventions were used to describe network structures:

K = linear activation function

L = log sigmoid, or 'logsig', activation function

T = tanh sigmoid, 'tansig', activation function

E.g. 6+K describes a simple linear network with 6 inputs. 7+2L+K describes a 7-input non-linear network with 2 'logsig' cells in a single hidden layer, with a linear output (scaling) cell.

Where pre-processing was investigated the abbreviations below indicate the method:

'm = n' Scaling through division by m, where m is the mean of the past n values

'$\div k$' Scaling through division by a constant.

E.g '$m = 2$', or '$\div 100$'

* Potentially optimum solutions in terms of accuracy and/or minimalist structure are indicated in the tables by an asterisk.

A3 Tables of typical experimental results – CPS CV prediction

A3.1 Linear networks

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0018	0.0179	0.0020	0
4	0.0019	0.0170	0.0020	0
5	0.0019	0.0169	0.0020	0
6	0.0019	0.0169	0.0020	0
7	0.0018	0.0170	0.0020	0
8	0.0018	0.0170	0.0020	1
9	0.0018	0.0162	0.0020	0
10	0.0018	0.0160	0.0020	0
11	0.0018	0.0155	0.0020	0
12	0.0018	0.0153	0.0020	0
*13	0.0018	0.0152	0.0020	0
14	0.0018	0.0152	0.0020	1
15	0.0018	0.0159	0.0020	0
16	0.0018	0.0158	0.0020	0
17	0.0018	0.0156	0.0020	1
18	0.0018	0.0158	0.0020	1
19	0.0018	0.0157	0.0020	0
20	0.0018	0.0156	0.0020	1

Table A3.1 No pre-processing of data

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0018	0.0152	0.0020	0
*4	0.0021	0.0139	0.0020	0
5	0.0019	0.0172	0.0020	1
6	0.0019	0.0169	0.0020	0
7	0.0018	0.0169	0.0020	0
8	0.0018	0.0171	0.0020	1
9	0.0018	0.0162	0.0020	1
10	0.0018	0.0162	0.0020	0
*11	0.0018	0.0149	0.0020	0
12	0.0018	0.0149	0.0020	0
13	0.0018	0.0152	0.0020	0
14	0.0018	0.0150	0.0020	1
15	0.0018	0.0159	0.0020	0
16	0.0018	0.0152	0.0020	1
17	0.0018	0.0156	0.0020	1
18	0.0018	0.0158	0.0020	1
19	0.0018	0.0153	0.0020	2
*20	0.0018	0.0147	0.0020	0

Table A3.2 Division by 100

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0008	0.0080	0.0011	0
4	0.0008	0.0081	0.0011	0
5	0.0008	0.0079	0.0010	0
6	0.0008	0.0082	0.0011	0
7	0.0008	0.0085	0.0010	1
8	0.0008	0.0077	0.0010	0
9	0.0008	0.0084	0.0010	2
10	0.0009	0.0075	0.0010	1
11	0.0009	0.0074	0.0010	1
*12	0.0009	0.0073	0.0010	1
13	0.0009	0.0075	0.0010	0
14	0.0009	0.0074	0.0010	0
15	0.0009	0.0076	0.0010	0
16	0.0009	0.0075	0.0010	0
17	0.0009	0.0074	0.0010	1
18	0.0009	0.0076	0.0010	1
19	0.0009	0.0075	0.0010	0
20	0.0009	0.0079	0.0010	0

Table A3.3 ("*m* = 2")

Inputs	*mean*	*max*	*s.d.*
3	0.0024	0.0217	0.0023
4	0.0024	0.0215	0.0023
5	0.0024	0.0213	0.0023
6	0.0023	0.0213	0.0023
*7	0.0023	0.0211	0.0022
8	0.0023	0.0216	0.0023
9	0.0023	0.0213	0.0023
10	0.0023	0.0219	0.0023
11	0.0023	0.0217	0.0023
12	0.0023	0.0217	0.0023
13	0.0023	0.0217	0.0023
14	0.0023	0.0216	0.0023
15	0.0023	0.0217	0.0022
16	0.0023	0.0217	0.0023
17	0.0023	0.0218	0.0023
18	0.0023	0.0217	0.0023
19	0.0023	0.0217	0.0023
20	0.0023	0.0218	0.0023

Table A3.3a post-processed

The maximum error ranges for each pre-processing method are shown in A3.4.

pre-proc?	*max error range (%)*
none	1.52 – 1.79
÷100	1.49 – 1.71
m = 2	0.73 – 0.85
	2.11 – 2.19**

** after post-processing

Table A3.4

Solutions which demonstrate low maximum error are indicated with and asterisk and are summarised below in table A3.5 (errors are expressed as percentages) :

preproc	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	13+1	0.18	1.52	0.2	0
÷ 100	11+1	0.18	1.49	0.2	0
÷ 100	20+1	0.18	1.47	0.20	0
m = 2	12+1	0.92	1.17	0.17	1
*m = 2***	7+K	0.23	2.11	0.22	1

**after post-processing

Table A3.5

The error values for the '*m=2*' solution are those obtained after post-processing of the results had taken place. Pre-processing by division by 100 offers optimum solutions, with the 11-input network appropriate for a minimalist implementation.

A3.2 Non-linear single-cell networks

Two experiments were performed on data pre-processed by division by 100 using both log sigmoid and tan sigmoid activation functions. Again 12 and 13-input networks offer viable solutions but do not achieve the accuracy attained by the linear solutions.

Inputs	*mean*	*max*	*s.d.*
3	0.0026	0.0175	0.0032
4	0.0026	0.0166	0.0032
5	0.0026	0.0167	0.0031
6	0.0026	0.0164	0.0031
7	0.0025	0.0170	0.0031
8	0.0025	0.0169	0.003
9	0.0025	0.0160	0.0029
10	0.0024	0.0158	0.0028
11	0.0024	0.0152	0.0028
*12	0.0024	0.0150	0.0027
13	0.0024	0.0151	0.0027
14	0.0024	0.0150	0.0027
15	0.0024	0.0156	0.0027
16	0.0023	0.0155	0.0026
17	0.0023	0.0154	0.0026
18	0.0023	0.0157	0.0026
19	0.0023	0.0156	0.0026
20	0.0023	0.0156	0.0025

Table A3.6 (log sigmoid)

inputs	*mean*	*max*	*s.d.*
3	0.0018	0.0180	0.0020
4	0.0019	0.0170	0.0020
5	0.0018	0.0170	0.0020
6	0.0018	0.0171	0.0020
7	0.0018	0.0170	0.0020
8	0.0018	0.0170	0.0020
9	0.0018	0.0159	0.0020
10	0.0018	0.0160	0.0020
11	0.0018	0.0156	0.0020
12	0.0018	0.0153	0.0020
*13	0.0018	0.0152	0.0020
14	0.0018	0.0153	0.0020
15	0.0018	0.0160	0.0020
16	0.0018	0.0159	0.0020
17	0.0018	0.0156	0.0020
18	0.0018	0.0154	0.0020
19	0.0018	0.0157	0.0020
20	0.0018	0.0160	0.0020

Table A3.7 (tanh sigmoid)

A3.3 Non-linear multi-cell networks

The above experiments in A3.2 were repeated with the number of inputs restricted to the range 11 – 13, as indicated by the linear network solutions, and with a single hidden whose cells were allowed to increase in number from 2 to 13 (in line with the number of inputs). Tables A3.8 – 13a show the first 10 results for the various network structures, sorted by maximum error. (In the case of the $m = 2$ pre-processed data, measurements were taken prior to post-processing.)

Without pre-processing:

inputs	*cells*	*mean*	*max*	*std*
*11	4	0.0018	0.0132	0.002
12	9	0.0028	0.0134	0.0025
12	11	0.0018	0.0143	0.002
11	3	0.0021	0.0147	0.0024
12	4	0.0019	0.0148	0.0021
12	2	0.0019	0.0149	0.0021
12	5	0.0019	0.0149	0.0021
*11	2	0.0019	0.0151	0.0021
11	6	0.0028	0.0151	0.0026
13	8	0.0025	0.0156	0.0028

Table A3.8 log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*12	3	0.0023	0.013	0.0023
12	5	0.0024	0.0132	0.0023
13	6	0.0021	0.0144	0.0023
12	6	0.0022	0.0146	0.0024
*12	2	0.0019	0.0148	0.0021
12	4	0.0019	0.0148	0.0021
12	8	0.0019	0.0148	0.0021
13	5	0.002	0.0149	0.0022
11	10	0.0023	0.0149	0.0024
13	2	0.0021	0.015	0.0024

Table A3.9 tanh sigmoid

Data divided by 100:

Inputs	*cells*	*mean*	*max*	*std*
*12	6	0.0023	0.0133	0.0023
12	5	0.0023	0.0146	0.0023
13	3	0.0021	0.0152	0.0022
11	3	0.0023	0.0161	0.0023
*12	2	0.0019	0.0171	0.0021
12	3	0.0023	0.0171	0.0024
13	11	0.0043	0.0172	0.0031
12	12	0.0040	0.0205	0.0035
11	8	0.0037	0.0225	0.0040
13	13	0.0045	0.0248	0.0036

Table A3.10 log sigmoid ÷100

Inputs	*cells*	*mean*	*Max*	*std*
*13	2	0.0020	0.0157	0.0020
13	7	0.0018	0.0163	0.0021
12	8	0.0027	0.0182	0.0033
13	3	0.0025	0.0187	0.0026
12	13	0.0028	0.0219	0.0028
12	12	0.0038	0.0219	0.0038
12	10	0.0050	0.0220	0.0036
11	6	0.0034	0.0221	0.0037
13	6	0.0036	0.0235	0.0039
12	3	0.0049	0.0262	0.0041

Table A3.11 tanh sig

Pre-processed using '$m = 2$' :

Inputs	*cells*	*mean*	*max*	*std*
*11	12	0.0009	0.0066	0.0010
11	7	0.0009	0.0068	0.0010
13	11	0.0009	0.0070	0.0010
11	11	0.0010	0.0070	0.0010
*12	3	0.0009	0.0071	0.0010
11	8	0.0009	0.0073	0.0010
12	10	0.0010	0.0073	0.0010
12	11	0.0010	0.0073	0.0010
12	12	0.0010	0.0073	0.0010
11	9	0.0010	0.0074	0.0010

Table A3.12 $m = 2$ logsig 2nd

Inputs	*cells*	*mean*	*Max*	*std*
*13	6	0.0026	0.0207	0.0026
13	5	0.0026	0.0209	0.0027
13	12	0.0024	0.0210	0.0024
13	9	0.0026	0.0211	0.0025
13	13	0.0023	0.0214	0.0024
13	11	0.0024	0.0216	0.0025
13	8	0.0025	0.0218	0.0025
*13	4	0.0025	0.0219	0.0026
13	7	0.0026	0.0225	0.0026
13	10	0.0023	0.0226	0.0025

Table A3.12a – after post-processing

Inputs	*cells*	*mean*	*max*	*std*
*11	9	0.0010	0.0066	0.0010
12	12	0.0010	0.0070	0.0010
11	2	0.0009	0.0071	0.0010
11	5	0.0009	0.0071	0.0010
13	5	0.0009	0.0071	0.0010
11	8	0.0010	0.0071	0.0010
11	12	0.0009	0.0072	0.0010
11	10	0.0010	0.0072	0.0010
12	3	0.0009	0.0073	0.0010
12	11	0.0010	0.0074	0.0010

Table A3.13 $m = 2$ tansig

Inputs	*cells*	*mean*	*max*	*std*
*13	8	0.0025	0.0199	0.0025
13	6	0.0023	0.0214	0.0024
13	11	0.0025	0.0214	0.0025
*13	3	0.0027	0.0217	0.0027
13	5	0.0024	0.0218	0.0024
13	7	0.0027	0.0224	0.0027
*13	2	0.0028	0.0225	0.0028
13	13	0.0026	0.0227	0.0025
13	4	0.0026	0.0229	0.0027
13	10	0.0025	0.0234	0.0025

Table A3.13a – after post-processing

pre-proc?	*function*	*max error range (%)*
none	log	1.32 – 20.5
	tanh	1.30 – 21.4
÷100	log	1.33 – 37.5
	tanh	1.57 – 10.1
$m = 2$**	Log	2.07 – 4.43
	Tanh	1.99 – 4.35

** after post-processing

Table A3.14 – maximum error ranges, 1 interval ahead

Table A3.14 summarises the above results in terms of maximum error ranges, while table A3.15 (overleaf) lists potential solutions selected from the above tables:

pre-proc?	*struct.*	*function*	*mean*	*max*	*s.d.*
none	11+4+K	log	0.18	1.32	0.20
	11+2+K		0.19	1.51	0.21
	12+3+K	tanh	0.23	1.30	0.23
100	12+6+K	log	0.23	1.33	0.23
	12+2+K		0.19	1.71	0.21
	13+2+K	tanh	0.20	1.57	0.20
$m = 2$	13+6+K	log	0.26	2.07	0.26
	13+4+K		0.25	2.19	0.26
	13+8+K	tanh	0.25	1.99	0.25
	13+2+K		0.28	2.25	0.28

Table A3.15 – potential solutions, 1 interval ahead

A3.4 Predicting two intervals ahead – linear networks

inputs	*mean*	*Max*	*s.d.*	*<0.01*
3	0.0030	0.0236	0.0026	0
4	0.0030	0.0228	0.0026	0
5	0.0030	0.0227	0.0026	0
6	0.0030	0.0227	0.0026	1
7	0.0029	0.0233	0.0026	0
8	0.0028	0.0232	0.0026	0
9	0.0028	0.0222	0.0025	0
10	0.0028	0.0217	0.0025	0
11	0.0028	0.0209	0.0025	1
12	0.0028	0.0209	0.0025	1
*13	0.0028	0.0206	0.0025	0
14	0.0028	0.0209	0.0025	0
15	0.0028	0.0212	0.0025	0
16	0.0028	0.0210	0.0025	0
17	0.0028	0.0212	0.0025	0
18	0.0028	0.0213	0.0025	0
19	0.0028	0.0212	0.0025	0
20	0.0028	0.0210	0.0025	0

Table A3.16 no pre-processing

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0029	0.0197	0.0026	0
4	0.0031	0.0224	0.0026	0
5	0.0030	0.0223	0.0026	0
6	0.0030	0.0227	0.0026	1
7	0.0029	0.0231	0.0026	0
8	0.0029	0.0230	0.0025	0
9	0.0029	0.0218	0.0025	0
10	0.0028	0.0217	0.0025	0
11	0.0028	0.0207	0.0025	0
*12	0.0028	0.0204	0.0025	0
13	0.0028	0.0206	0.0025	0
14	0.0028	0.0208	0.0025	0
15	0.0028	0.0212	0.0025	0
16	0.0028	0.021	0.0025	0
17	0.0028	0.0212	0.0025	0
18	0.0028	0.0212	0.0025	0
19	0.0028	0.0212	0.0025	1
20	0.0027	0.0205	0.0024	0

Table A3.17 ÷100

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0008	0.0084	0.00101	0
4	0.0008	0.0085	0.00101	0
5	0.0008	0.0087	0.00101	0
6	0.0008	0.0086	0.0010	1
7	0.0008	0.0083	0.0010	0
8	0.0008	0.0083	0.0010	0
9	0.0008	0.0081	0.0010	0
10	0.0008	0.0080	0.0010	1
*11	0.0008	0.0079	0.0010	1
12	0.0008	0.008	0.0010	1
13	0.0009	0.0081	0.0010	0
14	0.0009	0.0081	0.0010	0
15	0.0008	0.0081	0.0010	2
16	0.0008	0.0081	0.0010	0
17	0.0008	0.0082	0.0010	0
18	0.0008	0.0081	0.0010	1
19	0.0009	0.0083	0.0010	1
20	0.0008	0.0082	0.0010	2

Table A3.18 $m = 2$

inputs	*mean*	*Max*	*s.d.*
3	0.0037	0.0326	0.0034
4	0.0037	0.0324	0.0033
*5	0.0037	0.0322	0.0033
6	0.0037	0.0325	0.0033
7	0.0036	0.0326	0.0033
8	0.0036	0.0327	0.0033
9	0.0036	0.0327	0.0033
10	0.0036	0.0326	0.0033
11	0.0036	0.0326	0.0033
12	0.0036	0.0325	0.0033
13	0.0036	0.0325	0.0033
14	0.0036	0.0328	0.0033
15	0.0036	0.0327	0.0033
16	0.0036	0.0326	0.0033
17	0.0036	0.0326	0.0033
18	0.0036	0.0327	0.0033
19	0.0036	0.0324	0.0033
20	0.0036	0.0327	0.0033

Table A3.18a post-processed

pre-proc?	*max error range (%)*
none	2.06 – 2.36
÷100	1.97 – 2.30
$m = 2$**	3.22 – 3.28

** after post-processing

Table A3.19

preproc	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	13+K	0.28	2.06	0.25	0
÷ 100	3+K	0.29	1.97	0.26	0
÷ 100	12+K	0.28	2.04	0.25	0
$m = 2$	5+K	0.37	3.22	0.33	0

Table A3.20

Table A3.19 lists the maximum error ranges while table A3.20. summarises the above results, with those for $m = 2$ being adjusted for post-processing induced error. There is an increase of some 400% in the maximum error value. From the above tables non-linear solutions should require 11-13 inputs.

A3.5 Two intervals ahead – non-linear

inputs	cells	mean	max	Std
*11	13	0.0031	0.0172	0.0026
13	11	0.0037	0.0176	0.003
13	13	0.0033	0.0182	0.0026
11	8	0.0033	0.0194	0.0028
*12	4	0.0031	0.0203	0.0026
12	8	0.0035	0.0205	0.003
13	2	0.0029	0.0206	0.0026
13	3	0.0029	0.0206	0.0026
11	2	0.0032	0.0206	0.0029
12	3	0.0029	0.0209	0.0026

Table A3.21 log sigmoid

inputs	cells	mean	max	std
*12	6	0.0032	0.0174	0.0027
13	6	0.0036	0.0175	0.0032
11	6	0.0028	0.018	0.0024
*12	3	0.0031	0.0188	0.0027
13	5	0.0033	0.0189	0.0028
13	8	0.0036	0.0195	0.003
12	12	0.003	0.0201	0.0028
13	2	0.003	0.0204	0.0027
11	5	0.0031	0.0205	0.0028
12	4	0.003	0.0207	0.0027

Table A3.22 tanh sigmoid

A3.21 – 22 show results from unpre-processed data.

inputs	cells	mean	max	std
*12	7	0.004	0.0195	0.0031
13	3	0.0029	0.0208	0.0026
12	4	0.0032	0.0217	0.0028
*12	2	0.0031	0.0223	0.0027
13	4	0.0032	0.0229	0.0029
11	4	0.0045	0.0241	0.0042
12	3	0.0043	0.0242	0.0036
11	11	0.0045	0.0255	0.0039
13	6	0.0057	0.0268	0.0057
12	11	0.0054	0.0284	0.0048

Table A3.23 log sigmoid

inputs	cells	mean	max	std
*11	4	0.0035	0.0172	0.0031
13	3	0.003	0.0216	0.0026
13	7	0.0034	0.024	0.003
11	3	0.0043	0.0251	0.0048
*12	2	0.0037	0.0257	0.0034
13	8	0.0055	0.0262	0.0043
13	9	0.0047	0.0266	0.0048
13	5	0.0055	0.0291	0.0055
11	5	0.0052	0.0319	0.0055
12	6	0.0054	0.0323	0.0059

Table A3.24 tanh sigmoid

A3.23 – 4 show results for pre-processing via division by 100, while A3.25 – 6 below show those for $m = 2$.

inputs	cells	mean	max	std
*12	12	0.0010	0.0065	0.0010
13	10	0.0010	0.0065	0.00101
12	11	0.0010	0.0066	0.0010
*12	2	0.0009	0.0067	0.0010
11	11	0.0009	0.0068	0.0010
11	7	0.0010	0.0069	0.0010
11	2	0.0008	0.0071	0.0010
13	12	0.00101	0.0073	0.00101
11	13	0.0009	0.0074	0.0010
12	5	0.0010	0.0075	0.00101

Table A3.25 log sigmoid

inputs	cells	mean	max	std
*12	7	0.004	0.0311	0.0036
11	10	0.0041	0.0315	0.0036
*12	3	0.0037	0.0316	0.0034
13	13	0.0036	0.0318	0.0033
11	12	0.0037	0.0318	0.0033
13	4	0.0035	0.0319	0.0033
12	9	0.0036	0.0319	0.0033
11	9	0.0037	0.0319	0.0034
12	12	0.0036	0.0321	0.0033
13	3	0.004	0.0321	0.0036

Table A3.25a – after post-processing

inputs	cells	mean	max	std
*11	12	0.0010	0.0063	0.0010
12	8	0.0009	0.0069	0.00101
13	6	0.0010	0.0069	0.0010
13	8	0.0009	0.0071	0.0010
13	13	0.00101	0.0072	0.0010
*11	4	0.0008	0.0073	0.0010
11	5	0.0009	0.0073	0.0010
11	3	0.0008	0.0074	0.0010
13	3	0.0009	0.0074	0.0010
13	4	0.0009	0.0074	0.0010

Table A3.26 tanh sigmoid

inputs	cells	mean	max	std
*11	7	0.0039	0.0306	0.0035
11	11	0.0037	0.0316	0.0033
13	11	0.0036	0.0318	0.0034
11	6	0.0037	0.0318	0.0034
13	10	0.0038	0.032	0.0035
13	6	0.0037	0.0321	0.0033
11	5	0.0037	0.0322	0.0033
*12	3	0.004	0.0322	0.0037
13	5	0.004	0.0322	0.0036
12	5	0.0037	0.0323	0.0034

Table A3.26a – after post-processing

Table A3.27 summarises the above results in terms of maximum error ranges, while table A3.28 shows potential solutions.

pre-proc?	function	max error range (%)
None	log	1.72 – 20.5
	tanh	1.74– 2.84
÷100	log	1.95 – 12.19
	tanh	1.72 – 75.9
$m = 2$	log	3.11 – 3.52
	tanh	3.06 – 3.49

Table A3.27 – maximum error ranges, 2 intervals ahead

pre-proc?	Struct.	function	mean	max	s.d.
none	11+13+K	log	0.31	1.72	0.26
	12+4+K		0.31	2.03	0.26
	12+6+K	tanh	032	1.72	0.27
	12+3+K		0.31	1.88	0.27
÷100	12+7+K	log	0.40	1.95	0.31
	12+2+K		0.31	2.23	0.27
	11+4+K	tanh	0.35	1.72	0.31
	12+2+K		0.37	2.57	0.34
$m = 2$	12+7+K	log	0.40	3.11	0.36
	12+3+K		0.37	3.16	0.34
	11+7+K	tanh	0.39	3.06	0.35
	12+3+K		0.40	3.22	0.37

Table A3.28 – possible solutions, 2 intervals ahead

A4. HSM Phase 1

A4.1 Predicting CV

A4.1.1 1-interval ahead

Inputs	mean	max	s.d.	<0.01
*3	0.0057	0.1140	0.0071	0
4	0.0057	0.1140	0.0071	1
5	0.0057	0.1140	0.0071	0
6	0.0057	0.1141	0.0071	1
7	0.0057	0.1142	0.0071	1
8	0.0057	0.1142	0.0071	1
9	0.0057	0.1142	0.0071	3
10	0.0057	0.1142	0.0071	1
11	0.0057	0.1143	0.0071	1
12	0.0057	0.1143	0.0071	3
13	0.0057	0.1143	0.0071	1
14	0.0057	0.1143	0.0071	3
15	0.0057	0.1143	0.0071	3
16	0.0057	0.1143	0.0071	3
17	0.0057	0.1143	0.0071	3
18	0.0057	0.1143	0.0071	3
19	0.0057	0.1144	0.0071	6
20	0.0057	0.1144	0.0071	7

Table A4.1 unprocessed

inputs	mean	max	s.d.	<0.01
*3	0.0057	0.1140	0.0071	0
4	0.0057	0.1140	0.0071	1
5	0.0057	0.1140	0.0071	0
6	0.0057	0.1141	0.0071	1
7	0.0057	0.1142	0.0071	1
8	0.0057	0.1142	0.0071	1
9	0.0057	0.1142	0.0071	3
10	0.0057	0.1142	0.0071	2
11	0.0057	0.1143	0.0071	1
12	0.0057	0.1143	0.0071	3
13	0.0057	0.1143	0.0071	1
14	0.0057	0.1143	0.0071	3
15	0.0057	0.1143	0.0071	3
16	0.0057	0.1143	0.0071	3
17	0.0057	0.1143	0.0071	3
18	0.0057	0.1143	0.0071	4
19	0.0057	0.1144	0.0071	4
20	0.0057	0.1144	0.0071	8

Table A4.2 – ÷100

inputs	mean	max	s.d.	<0.01
3	0.0028	0.0579	0.0036	0
4	0.0028	0.0579	0.0036	1
5	0.0028	0.0579	0.0036	0
6	0.0028	0.0579	0.0036	0
7	0.0028	0.0579	0.0036	0
8	0.0028	0.0579	0.0036	0
9	0.0028	0.0579	0.0036	1
10	0.0028	0.0579	0.0036	0
11	0.0028	0.0579	0.0036	0
12	0.0028	0.0579	0.0036	0
13	0.0028	0.0579	0.0036	0
14	0.0028	0.0579	0.0036	0
15	0.0028	0.0579	0.0036	1
16	0.0028	0.0579	0.0036	1
17	0.0028	0.0579	0.0036	2
18	0.0028	0.0579	0.0036	1
19	0.0028	0.0579	0.0036	1
20	0.0028	0.0579	0.0036	1

Table A4.3 – '*m* = 2'

inputs	mean	max	s.d.
3	0.0070	0.1228	0.0086
*4	0.0070	0.1227	0.0086
5	0.0070	0.1229	0.0086
6	0.0070	0.1231	0.0086
7	0.0070	0.1231	0.0086
8	0.0070	0.1232	0.0086
9	0.0070	0.1233	0.0086
10	0.0070	0.1233	0.0086
11	0.0070	0.1234	0.0086
12	0.0070	0.1234	0.0086
13	0.0070	0.1234	0.0086
14	0.0070	0.1234	0.0086
15	0.0070	0.1234	0.0086
16	0.0070	0.1234	0.0086
17	0.0070	0.1234	0.0086
18	0.0070	0.1234	0.0086
19	0.0070	0.1234	0.0086
20	0.0070	0.1234	0.0086

Table A4.3a – post-processed

A4.1.2 2-intervals ahead

inputs	*mean*	*Max*	*s.d.*	*<0.01*
*3	0.0097	0.1678	0.0111	0
4	0.0097	0.1678	0.0111	0
5	0.0097	0.1682	0.0112	0
6	0.0097	0.1687	0.0112	1
7	0.0096	0.1689	0.0112	0
8	0.0096	0.1690	0.0112	0
9	0.0097	0.1694	0.0112	0
10	0.0097	0.1695	0.0112	1
11	0.0097	0.1696	0.0112	2
12	0.0096	0.1697	0.0112	2
13	0.0097	0.1698	0.0112	3
14	0.0096	0.1699	0.0112	4
15	0.0096	0.1699	0.0112	5
16	0.0097	0.1699	0.0112	5
17	0.0097	0.1699	0.0112	4
18	0.0097	0.1699	0.0112	4
19	0.0097	0.1699	0.0112	6
20	0.0097	0.1699	0.0112	6

Table A4.4 unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0097	0.1678	0.0111	0
4	0.0097	0.1678	0.0111	0
5	0.0097	0.1682	0.0112	0
6	0.0097	0.1687	0.0112	1
7	0.0096	0.1689	0.0112	0
8	0.0096	0.169	0.0112	0
9	0.0097	0.1694	0.0112	0
10	0.0097	0.1695	0.0112	1
11	0.0097	0.1696	0.0112	2
12	0.0096	0.1697	0.0112	2
13	0.0097	0.1698	0.0112	3
14	0.0096	0.1699	0.0112	4
15	0.0096	0.1699	0.0112	5
16	0.0097	0.1699	0.0112	5
17	0.0097	0.1699	0.0112	4
18	0.0097	0.1699	0.0112	4
19	0.0097	0.1699	0.0112	6
20	0.0097	0.1699	0.0112	8

Table A4.5 – ÷100

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.003	0.0579	0.0037	0
4	0.0029	0.0579	0.0037	0
5	0.0029	0.0579	0.0037	0
6	0.0029	0.0579	0.0037	0
7	0.0029	0.0579	0.0037	0
8	0.003	0.0579	0.0037	0
9	0.003	0.0579	0.0037	0
10	0.003	0.0579	0.0037	0
11	0.003	0.0579	0.0037	0
12	0.003	0.0579	0.0037	0
13	0.003	0.0579	0.0037	0
14	0.003	0.0579	0.0037	1
15	0.003	0.0579	0.0037	1
16	0.003	0.0579	0.0037	2
17	0.003	0.0579	0.0037	1
18	0.003	0.0579	0.0037	1
19	0.003	0.0579	0.0037	1
20	0.003	0.0579	0.0037	1

Table A4.6 – '*m* = 2'

inputs	*mean*	*max*	*s.d.*
*3	0.0104	0.1774	0.0126
4	0.0104	0.1776	0.0126
5	0.0104	0.1775	0.0126
6	0.0104	0.1775	0.0126
7	0.0104	0.1776	0.0126
8	0.0104	0.1776	0.0126
9	0.0104	0.1775	0.0126
10	0.0104	0.1775	0.0126
11	0.0104	0.1776	0.0126
12	0.0104	0.1775	0.0126
13	0.0104	0.1776	0.0126
14	0.0104	0.1776	0.0126
15	0.0104	0.1775	0.0125
16	0.0104	0.1775	0.0125
17	0.0104	0.1775	0.0125
18	0.0104	0.1774	0.0125
19	0.0104	0.1774	0.0125
20	0.0104	0.1774	0.0125

Table A4.6a – post-processed

The tables below summarise the above results and those from the previous section. There is minimal difference between the maximum error ranges with the unprocessed training data networks and those using the 'division by 100' method being almost identical.

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	11.40 – 11.43	16.78 – 16.99
÷100	11.40 – 11.44	16.78 – 16.99
$m = 2$	5.79 (all)	5.79 (all)
	12.27 – 12.34**	17.74 – 17.76

** after post-processing

Table A4.7 – maximum error ranges

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*Mean*	*max*	*s.d.*	*<0.01*
none	3+K	0.57	11.40	0.71	0	3+K	0.97	16.78	1.11	0
÷ 100	3+K	0.57	11.40	0.71	0	3+K	0.97	16.78	1.11	0
$m = 2$**	4+K	0.70	12.27	0.86	1	3+K	1.04	17.74	1.26	0

**after post-processing

Table A4.8 – possible network solutions

It should be noted that the '$m = 2$' method shows fewer potentially extraneous inputs in the overall results tables than the other forms of pre-processing but it is significantly less accurate in all accuracy measurement parameters.

A4.1.3 Predicting CV using CO and H_2 as additional input parameters (1-interval ahead)

Inputs (CV+CO)	*mean*	*max*	*s.d.*	*<0.01**
3+1	0.0058	0.1143	0.0070	0
3+2	0.0057	0.1142	0.0070	0
3+3	0.0058	0.1143	0.0070	0
3+4	0.0057	0.1142	0.0070	0
3+5	0.0057	0.1142	0.0070	0
3+6	0.0057	0.1142	0.0070	0
4+1	0.0058	0.1142	0.0070	1
4+2	0.0057	0.1142	0.0070	1
4+3	0.0058	0.1143	0.0070	1
4+4	0.0057	0.1142	0.0070	1
4+5	0.0057	0.1142	0.0070	1
4+6	0.0057	0.1142	0.0070	1
5+1	0.0057	0.1143	0.0070	2
5+2	0.0057	0.1143	0.0070	2
5+3	0.0058	0.1143	0.0070	2
5+4	0.0057	0.1142	0.0070	2
5+5	0.0057	0.1142	0.0070	2
5+6	0.0057	0.1142	0.0070	2
6+1	0.0057	0.1143	0.0070	3
6+2	0.0057	0.1143	0.0070	3
6+3	0.0057	0.1143	0.0070	3
6+4	0.0057	0.1143	0.0070	3
6+5	0.0057	0.1143	0.0070	3
6+6	0.0057	0.1143	0.0070	3

Table A4.9 additional CO input

Inputs ($CV+H_2$)	*mean*	*max*	*s.d.*	*<0.01**
3+1	0.0058	0.1141	0.0070	0
3+2	0.0058	0.1141	0.0070	0
3+3	0.0058	0.1142	0.0070	0
3+4	0.0057	0.1141	0.0070	0
3+5	0.0057	0.1141	0.0070	0
3+6	0.0057	0.1141	0.0070	0
4+1	0.0058	0.1141	0.0070	1
4+2	0.0058	0.1141	0.0070	1
4+3	0.0058	0.1142	0.0070	1
4+4	0.0057	0.1141	0.0070	1
4+5	0.0057	0.1141	0.0070	1
4+6	0.0057	0.1141	0.0070	1
5+1	0.0058	0.1142	0.0070	2
5+2	0.0058	0.1142	0.0070	2
5+3	0.0058	0.1142	0.0070	2
5+4	0.0057	0.1142	0.0070	2
5+5	0.0057	0.1142	0.0070	2
5+6	0.0057	0.1141	0.0070	2
6+1	0.0057	0.1142	0.0070	3
6+2	0.0057	0.1142	0.0070	3
6+3	0.0058	0.1142	0.0070	3
6+4	0.0057	0.1142	0.0070	3
6+5	0.0057	0.1142	0.0070	3
6+6	0.0057	0.1142	0.0070	3

Table A4.10 additional H_2 input

*The pattern (and number) of potentially extraneous inputs was identical for both parameters in each case.

Table A4.9 shows the results for a linear network investigation using an additional input parameter of percentage CO content, and without pre-processing, whilst Table A4.10 shows an identical investigation for Hydrogen. The network was allowed in both cases to expand from 3 inputs to a maximum of 6, with those for CO and H_2 increasing from 1 to 6.

The training data set for that value used decimal fractions – i.e. values less than 1 for the percentage values, hence no pre-processing was required The results are comparable with those obtained with a single CV input parameter and suggest that tracking additional parameters has negligible effect on accuracy.

A4.2 Predicting specific gravity (s.g.)

All data was in the range < 1.0; thus no pre-processing was required

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0061	0.1806	0.0082	0
4	0.0061	0.1806	0.0082	1
5	0.0061	0.1806	0.0082	0
6	0.0061	0.1806	0.0082	0
*7	0.0061	0.1806	0.0081	1
8	0.0061	0.1806	0.0082	2
9	0.0061	0.1806	0.0082	2
10	0.0061	0.1806	0.0081	0
11	0.0061	0.1806	0.0082	2
12	0.0061	0.1806	0.0081	2
13	0.0061	0.1806	0.0081	3
14	0.0061	0.1806	0.0082	2
15	0.0061	0.1807	0.0082	3
16	0.0061	0.1807	0.0082	3
17	0.0061	0.1807	0.0081	2
18	0.0061	0.1807	0.0081	3
19	0.0061	0.1807	0.0081	5
20	0.0061	0.1807	0.0081	5

Table 6.11 1-interval ahead

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0096	0.1948	0.0117	0
4	0.0096	0.1948	0.0117	0
5	0.0096	0.1948	0.0117	0
*6	0.0096	0.1947	0.0117	0
7	0.0096	0.1947	0.0117	0
8	0.0096	0.1947	0.0117	1
9	0.0096	0.1948	0.0117	1
10	0.0096	0.1948	0.0117	1
11	0.0096	0.1949	0.0117	1
12	0.0096	0.1948	0.0117	1
13	0.0096	0.1949	0.0117	1
14	0.0096	0.1949	0.0117	1
15	0.0096	0.1949	0.0117	1
16	0.0096	0.1949	0.0117	1
17	0.0096	0.1950	0.0117	2
18	0.0096	0.1950	0.0117	2
19	0.0096	0.1950	0.0117	3
20	0.0096	0.1950	0.0117	4

Table A4.12 2-intervals ahead

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	18.06 – 18.07	19.47 – 19.50

Table A4.13 – maximum error ranges

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*Mean*	*max*	*s.d.*	*<0.01*
None	7+K**	0.61	18.06	0.81	1	6+K	0.96	19.47	1.17	0

** selected for lower s.d.

Table A4.14 – possible network solutions

Note that there are fewer potentially extraneous inputs in the results for two intervals ahead solutions than those for a single interval ahead.

A4.3 Predicting air/fuel ratio (a/f)

A4.3.1 I sampling interval ahead

Inputs	mean	max	s.d.	<0.01
3	0.0097	0.1308	0.0111	0
*4	0.0097	0.1307	0.0111	1
5	0.0097	0.1309	0.0111	0
6	0.0097	0.1308	0.011	1
7	0.0097	0.1309	0.011	0
8	0.0097	0.1309	0.011	2
9	0.0097	0.1311	0.011	1
10	0.0097	0.1314	0.011	2
11	0.0097	0.1314	0.011	3
12	0.0097	0.1314	0.011	4
13	0.0097	0.1314	0.011	3
14	0.0097	0.1314	0.011	4
15	0.0097	0.1315	0.011	5
16	0.0097	0.1314	0.011	4
17	0.0097	0.1314	0.011	5
18	0.0097	0.1314	0.011	6
19	0.0097	0.1314	0.011	7
20	0.0097	0.1314	0.011	6

Table A4.15 unprocessed

inputs	mean	max	s.d.	<0.01
3	0.0097	0.1308	0.0111	0
*4		0.1307	0.0111	1
5	0.0097	0.1309	0.0111	0
6	0.0097	0.1308	0.0110	1
7	0.0097	0.1309	0.0110	0
8	0.0097	0.1309	0.0110	2
9	0.0097	0.1311	0.0110	1
10	0.0097	0.1314	0.0110	2
11	0.0097	0.1314	0.0110	3
12	0.0097	0.1314	0.0110	4
13	0.0097	0.1314	0.0110	3
14	0.0097	0.1315	0.0110	7
15	0.0097	0.1315	0.0110	5
16	0.0097	0.1314	0.0110	4
17	0.0097	0.1314	0.0110	5
18	0.0097	0.1314	0.0110	6
19	0.0097	0.1314	0.0110	7
20	0.0097	0.1314	0.0110	6

Table A4.16 '÷ 100'

A4.3.2 2 sampling intervals ahead

inputs	mean	max	s.d.	<0.01
3	0.016	0.1891	0.0167	0
4	0.016	0.1893	0.0167	0
*5	0.016	0.1888	0.0167	0
6	0.016	0.1890	0.0167	1
7	0.016	0.1890	0.0167	0
8	0.016	0.1894	0.0167	2
9	0.016	0.1897	0.0167	2
10	0.016	0.1899	0.0167	4
11	0.016	0.1900	0.0167	4
12	0.016	0.1900	0.0167	5
13	0.016	0.1900	0.0167	6
14	0.016	0.1900	0.0167	8
15	0.016	0.1899	0.0167	9
16	0.016	0.1899	0.0167	7
17	0.016	0.1899	0.0167	7
18	0.016	0.1899	0.0167	7
19	0.016	0.1899	0.0167	8
20	0.016	0.1899	0.0167	8

Table A4.17 unprocessed

inputs	mean	max	s.d.	<0.01
3	0.016	0.1891	0.0167	0
4	0.016	0.1893	0.0167	0
*5	0.016	0.1888	0.0167	0
6	0.016	0.1890	0.0167	1
7	0.016	0.1890	0.0167	0
8	0.016	0.1894	0.0167	2
9	0.016	0.1897	0.0167	2
10	0.016	0.1899	0.0167	4
11	0.016	0.1900	0.0167	4
12	0.016	0.1900	0.0167	5
13	0.016	0.1899	0.0167	1
14	0.016	0.1900	0.0167	8
15	0.016	0.1899	0.0167	9
16	0.016	0.1899	0.0167	7
17	0.016	0.1899	0.0167	7
18	0.016	0.1899	0.0167	7
19	0.016	0.1899	0.0167	9
20	0.016	0.1899	0.0167	8

Table A4.18 '÷ 100'

The two methods of pre-processing produced near identical results, which are summarised in the tables below.

max error range (%)	
1 interval ahead	*2 intervals ahead*
13.078 – 13.14	18.88 – 19.00

Table A4.19 – maximum error ranges

1 interval ahead					*2 intervals ahead*				
Struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
4+K	0.97	13.07	1.11	1	5+K	1.60	18.88	1.67	0

** selected for lower s.d.

Table A4.20 – possible network solutions

Of note is the steady increase in the number of potentially extraneous inputs for networks where the number of inputs exceeds seven.

A4.4 Predicting Wobbe indices

A4.4.1 1 sampling interval ahead

inputs	*Mean*	*max*	*s.d.*	*<0.01*
3	0.0088	0.1258	0.0095	0
4	0.0088	0.1252	0.0095	0
5	0.0088	0.1250	0.0095	0
6	0.0087	0.1246	0.0095	2
7	0.0087	0.1220	0.0095	0
8	0.0087	0.1222	0.0095	0
9	0.0087	0.1218	0.0095	0
10	0.0087	0.1211	0.0095	0
11	0.0087	0.1205	0.0095	1
12	0.0087	0.1213	0.0095	2
13	0.0087	0.1208	0.0095	1
14	0.0087	0.1208	0.0095	3
15	0.0087	0.1204	0.0095	3
16	0.0087	0.1204	0.0095	2
17	0.0087	0.1208	0.0095	3
*18	0.0087	0.1187	0.0095	3
19	0.0087	0.1191	0.0095	3
20	0.0087	0.1197	0.0095	4

Table A4.21 unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0088	0.1258	0.0095	0
4	0.0088	0.1252	0.0095	0
5	0.0088	0.1250	0.0095	0
6	0.0087	0.1246	0.0095	2
7	0.0087	0.1220	0.0095	0
8	0.0087	0.1222	0.0095	0
9	0.0087	0.1218	0.0095	0
10	0.0087	0.1211	0.0095	0
11	0.0087	0.1205	0.0095	1
12	0.0087	0.1213	0.0095	2
13	0.0087	0.1208	0.0095	1
14	0.0087	0.1208	0.0095	2
15	0.0087	0.1204	0.0095	3
16	0.0087	0.1204	0.0095	2
17	0.0087	0.1208	0.0095	3
*18	0.0087	0.1187	0.0095	3
19	0.0087	0.1191	0.0095	3
20	0.0087	0.1197	0.0095	4

Table A4.22 '÷ 100'

A4.4.2 2 sampling intervals ahead

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0138	0.1450	0.0139	0
4	0.0137	0.1404	0.0139	0
5	0.0137	0.1393	0.0139	0
6	0.0136	0.1419	0.0139	0
7	0.0136	0.1415	0.0139	2
8	0.0136	0.1414	0.0139	3
9	0.0136	0.1413	0.0139	4
10	0.0135	0.1403	0.0139	3
*11	0.0135	0.1393	0.0139	4
12	0.0135	0.1395	0.0139	5
13	0.0135	0.1402	0.0139	4
14	0.0135	0.1404	0.0139	5
15	0.0135	0.1407	0.0139	6
16	0.0135	0.1413	0.0139	4
17	0.0135	0.1443	0.0139	7
18	0.0135	0.1459	0.0139	7
19	0.0135	0.1462	0.0139	6
20	0.0135	0.1463	0.0139	6

Table A4.23 unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0138	0.1450	0.0139	0
4	0.0137	0.1404	0.0139	0
5	0.0137	0.1393	0.0139	0
6	0.0136	0.1419	0.0139	0
7	0.0136	0.1415	0.0139	2
8	0.0136	0.1414	0.0139	3
9	0.0136	0.1413	0.0139	4
10	0.0135	0.1403	0.0139	3
*11	0.0135	0.1394	0.0139	4
12	0.0135	0.1395	0.0139	5
13	0.0135	0.1402	0.0139	4
14	0.0135	0.1404	0.0139	5
15	0.0135	0.1407	0.0139	7
16	0.0135	0.1413	0.0139	4
17	0.0135	0.1443	0.0139	7
18	0.0135	0.1459	0.0139	7
19	0.0135	0.1462	0.0139	6
20	0.0135	0.1463	0.0139	6

Table A4.24 '÷ 100'

Again the two methods of pre-processing produced near identical results, which are summarised in the tables below.

max error range (%)	
1 interval ahead	*2 intervals ahead*
11.87 – 12.58	13.93 – 14.63

Table A4.25 – maximum error ranges

1 interval ahead					*2 intervals ahead*				
Struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
18+K	0.87	11.87	0.95	3	11+K	1.35	13.93	1.39	4

Table A4.26 – possible network solutions

Again there is a steady increase in the number of potentially extraneous inputs for networks where the number of inputs exceeds seven.

A5 Dawes Lane Coke Ovens

A5.1 Predicting CV

A5.1.1 Predicting one interval ahead – linear networks

inputs	mean	max	s.d.	<0.01
3	0.003	0.0393	0.0034	0
*4	0.0029	0.0391	0.0034	0
5	0.0029	0.0398	0.0034	0
6	0.0028	0.0401	0.0031	0
7	0.0028	0.0407	0.0032	0
8	0.0027	0.0411	0.0031	0
9	0.0027	0.041	0.0031	0
10	0.0027	0.0414	0.0031	0
11	0.0027	0.0414	0.0032	1
12	0.0027	0.0415	0.0032	0
13	0.0027	0.0413	0.0032	0
14	0.0027	0.0411	0.0032	1
15	0.0027	0.041	0.0032	0
16	0.0027	0.041	0.0032	1
17	0.0027	0.041	0.0032	0
18	0.0027	0.041	0.0032	0
19	0.0027	0.041	0.0032	0
20	0.0027	0.0411	0.0032	1

Table A5.1 unprocessed data

Inputs	mean	max	s.d.	<0.01
3	0.0030	0.0393	0.0034	0
*4	0.0029	0.0391	0.0034	0
5	0.0029	0.0398	0.0034	0
6	0.0028	0.0401	0.0031	0
7	0.0028	0.0407	0.0032	1
8	0.0027	0.0411	0.0031	0
9	0.0027	0.041	0.0031	0
10	0.0027	0.0414	0.0031	0
11	0.0027	0.0414	0.0032	1
12	0.0027	0.0415	0.0032	0
13	0.0027	0.0413	0.0032	0
14	0.0027	0.0412	0.0031	0
15	0.0027	0.0411	0.0032	1
16	0.0027	0.041	0.0032	1
17	0.0027	0.041	0.0032	0
18	0.0027	0.041	0.0032	0
19	0.0027	0.041	0.0032	0
20	0.0027	0.0411	0.0032	1

Table A5.2 division by 100

inputs	mean	max	s.d.	<0.01
3	0.0016	0.0208	0.0018	0
*4	0.0015	0.0207	0.0016	0
5	0.0015	0.0209	0.0016	0
6	0.0014	0.021	0.0016	0
7	0.0014	0.021	0.0016	0
8	0.0014	0.0211	0.0016	0
9	0.0014	0.0212	0.0016	1
10	0.0014	0.0212	0.0016	0
11	0.0014	0.0211	0.0016	1
12	0.0014	0.021	0.0016	1
13	0.0014	0.0209	0.0016	0
14	0.0014	0.0209	0.0016	1
15	0.0014	0.0209	0.0016	1
16	0.0014	0.0209	0.0016	2
17	0.0014	0.0209	0.0016	1
18	0.0014	0.021	0.0016	0
19	0.0014	0.021	0.0016	2
20	0.0014	0.0209	0.0016	1

Table A5.3 '*m* = 2'

inputs	mean	max	s.d.
*3	0.0044	0.0424	0.0047
4	0.0043	0.0427	0.0046
5	0.0043	0.0428	0.0046
6	0.0044	0.0431	0.0045
7	0.0043	0.0429	0.0045
8	0.0044	0.0431	0.0045
9	0.0044	0.0432	0.0045
10	0.0044	0.0431	0.0045
11	0.0044	0.0432	0.0045
12	0.0044	0.0431	0.0045
13	0.0044	0.0431	0.0045
14	0.0044	0.043	0.0045
15	0.0044	0.0431	0.0045
16	0.0044	0.043	0.0045
17	0.0045	0.043	0.0045
18	0.0045	0.043	0.0045
19	0.0044	0.043	0.0045
20	0.0045	0.043	0.0045

Table A5.3a – after post-processing

The maximum error ranges for each pre-processing method are shown in 7.1.2.4.

p.p.	*max error range (%)*
none	3.91 – 4.15
÷100	3.91 – 4.15
m = 2	2.07 – 2.12
	4.24 – 4.32**

** after post-processing

Table A5.4

Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in Table A5.1.2.5 (errors are expressed as percentages) :

p.p	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	4+K	0.30	3.93	0.34	0
÷ 100	4+K	0.29	3.91	0.34	0
m = 2	4+K	0.15	2.07	0.16	0
m = 2**	3+K	0.44	4.24	0.47	0

**after post-processing

Table A5.5

There is little difference between the network trained without out pre-processing and that using '÷ 100' data. However their maximum error levels are now approaching that of the '*m* = 2' solution, although the latter displays lower accuracy in terms of mean absolute error and standard deviation.

A5.1.2 Predicting one interval ahead (non-linear networks)

There is little in the above tables to give adequate information on the number of inputs to be used; the author is sceptical of the 3-4 input results in that they infer that only knowledge of the past 3 minutes is sufficient to produce a an accurate prediction over a longer operating period. However, further examination of the above tables shows that whilst maximum error values rise after the low-input network solutions there is a fall in error levels for 14 – 19 input networks. Accordingly it was decided to seek solutions using both 3 – 6 input networks and those with 14 – 19 inputs. In each case the hidden layer would be allowed to grow from 3 to 20 cells. However stability problems with the software lead to a revision of strategy: for the smaller networks it was established that as the number of cells approached the number of inputs 'singular matrix' warnings were encountered and the test had to be terminated. For the larger networks the number of cells had to be limited to some 4 to 6 depending on the number of inputs; it was possible to increase the number of cells by decreasing the number of inputs.

Networks with 3 to 6 inputs:

inputs	cells	mean	max	s.d.
*5	2	0.0124	0.0329	0.0089
5	4	0.0124	0.0329	0.0089
5	3	0.0121	0.0333	0.0087
4	4	0.0028	0.0366	0.0035
4	3	0.0029	0.037	0.0033
4	5	0.0028	0.0371	0.0035
3	4	0.0029	0.0375	0.0033
3	3	0.003	0.0386	0.0034
3	5	0.0029	0.039	0.0033
4	2	0.0029	0.0391	0.0034
3	2	0.003	0.0394	0.0034
6	5	0.0025	0.0397	0.003
6	2	0.0028	0.04	0.0031
6	3	0.0023	0.0407	0.003
6	4	0.0028	0.0407	0.0032
5	5	0.0028	0.0431	0.0039

Table A5.6 unprocessed data – log sigmoid
(No. of cells restricted to 5)

inputs	cells	mean	max	s.d.
*4	2	0.0029	0.0391	0.0034
5	2	0.0028	0.0394	0.0035
3	3	0.0031	0.0394	0.0034
5	3	0.0029	0.0396	0.0034
4	3	0.003	0.0396	0.0035
3	2	0.003	0.0399	0.0035

Table A5.7 unprocessed data – tanh sigmoid.
(No. of cells restricted to 3)

inputs	cells	mean	max	std
*4	6	0.0029	0.0359	0.0033
3	4	0.0029	0.0362	0.0032
3	3	0.0029	0.0373	0.0034
3	5	0.0029	0.0373	0.0035
4	4	0.0029	0.0374	0.0035
*6	2	0.0028	0.0375	0.0037
5	5	0.0026	0.0376	0.0032
5	3	0.0029	0.0377	0.0035
4	2	0.0029	0.038	0.0034
4	5	0.0028	0.0381	0.0034
6	5	0.0026	0.0384	0.0033
3	2	0.003	0.0387	0.0034
6	6	0.0026	0.0391	0.0032
6	4	0.0025	0.0393	0.0036
4	3	0.003	0.0393	0.0034
5	2	0.0029	0.0395	0.0033
5	4	0.0056	0.0399	0.0046
6	3	0.0041	0.041	0.004
5	6	0.0029	0.0453	0.0038
3	6	0.0032	0.0759	0.0051

Table A5.8 '÷100' – log sigmoid
(No. of cells restricted to 6)

inputs	cells	mean	max	std
*4	4	0.0029	0.0362	0.0037
4	5	0.0027	0.0368	0.0035
3	5	0.0029	0.0374	0.0035
5	5	0.0027	0.0378	0.0032
5	6	0.0027	0.0378	0.0034
3	3	0.003	0.0379	0.0033
3	6	0.0031	0.0383	0.0034
4	6	0.003	0.0384	0.0034
4	3	0.0041	0.0384	0.0039
5	4	0.0029	0.0387	0.0034
*3	2	0.0029	0.0389	0.0037
4	2	0.003	0.0389	0.0034
6	5	0.0025	0.0391	0.0033
6	3	0.0029	0.0391	0.0031
6	2	0.0029	0.0392	0.0031
6	4	0.0024	0.0398	0.0037
5	2	0.0029	0.0402	0.0032
5	3	0.0028	0.0422	0.0039
3	4	0.0032	0.0495	0.0039
6	6	0.0031	0.067	0.0043

Table A5.9 '÷100' – tanh sigmoid
(No. of cells restricted to 6)

inputs	*cells*	*mean*	*max*	*std*
4	4	0.0011	0.0195	0.0015
3	3	0.0012	0.0195	0.0016
6	5	0.0012	0.0195	0.0014
3	4	0.0014	0.0195	0.0017
3	5	0.0013	0.0198	0.0016
3	2	0.0014	0.02	0.0017
5	5	0.0014	0.0201	0.0017
5	4	0.0012	0.0203	0.0014
4	3	0.0014	0.0203	0.0015
4	5	0.0014	0.0203	0.0016
5	3	0.0015	0.0203	0.0017
4	2	0.0016	0.021	0.0021
5	2	0.0016	0.0248	0.0021
6	4	0.0016	0.0277	0.0021
6	3	0.0014	0.0447	0.0025
6	2	0.0018	0.0679	0.0049

Table A5.10 – '*m* = 2' – log sigmoid
(No. of cells restricted to 5)

inputs	*cells*	*mean*	*max*	*std*
*4	4	0.0034	0.0414	0.0039
6	5	0.0034	0.0414	0.0038
3	3	0.0035	0.0414	0.004
3	4	0.0036	0.0414	0.0041
3	5	0.0035	0.0418	0.0039
*3	2	0.0036	0.042	0.0041
5	5	0.0035	0.0421	0.0039
4	2	0.0038	0.0421	0.0043
4	5	0.0034	0.0422	0.004
4	3	0.0035	0.0422	0.0039
5	2	0.004	0.0422	0.0046
5	4	0.0034	0.0423	0.0038
5	3	0.0037	0.0423	0.0041
6	4	0.0038	0.0426	0.0042
6	3	0.0036	0.0459	0.0043
6	2	0.0041	0.069	0.0061

Table A5.10a – post-processed

inputs	*cells*	*mean*	*max*	*std*
5	5	0.0011	0.0196	0.0015
6	4	0.0012	0.0198	0.0015
4	3	0.0016	0.0201	0.002
3	5	0.0014	0.0202	0.0017
5	4	0.0011	0.0205	0.0016
6	2	0.0012	0.0206	0.0015
4	2	0.0013	0.0206	0.0016
4	4	0.0014	0.0272	0.0023
4	5	0.0013	0.0279	0.0022
5	2	0.0013	0.0347	0.0026
3	2	0.0015	0.0551	0.0029
6	5	0.0015	0.0977	0.006
3	4	0.0017	0.1163	0.0053
5	3	0.0015	0.122	0.0056
3	3	0.0021	0.1516	0.0082
6	3	0.0017	0.1649	0.0073

Table A5.11 – '*m* = 2' – tanh sigmoid
(No. of cells restricted to 5)

inputs	*cells*	*mean*	*max*	*std*
*5	5	0.0033	0.0415	0.0037
6	4	0.0034	0.0417	0.0038
4	4	0.0036	0.0417	0.0045
*5	2	0.0035	0.042	0.0044
3	5	0.0037	0.0421	0.0042
4	3	0.004	0.0421	0.0045
5	4	0.0033	0.0424	0.0039
4	2	0.0034	0.0425	0.0039
6	2	0.0033	0.0426	0.0038
4	5	0.0035	0.0481	0.0045
3	2	0.0038	0.0563	0.0048
3	4	0.0038	0.0989	0.0059
6	5	0.0037	0.1192	0.0075
5	3	0.0037	0.1231	0.0065
3	3	0.0044	0.148	0.0086
6	3	0.0039	0.1662	0.0081

Table A5.11a – post-processed

Networks with 14 to 17 inputs:

inputs	cells	mean	max	std
*17	3	0.0124	0.033	0.0091
14	3	0.0125	0.0333	0.0091
*14	2	0.0099	0.0374	0.0072
15	5	0.0026	0.0387	0.0033
14	5	0.0026	0.0389	0.0032
16	2	0.0027	0.039	0.0032
17	5	0.0027	0.0393	0.0035
16	5	0.0027	0.0394	0.0031
14	4	0.0028	0.0398	0.0032
15	3	0.0027	0.0402	0.0035
16	4	0.0029	0.0407	0.0035
17	4	0.0027	0.0408	0.0032
17	2	0.0027	0.041	0.0032
15	4	0.0029	0.0411	0.0033
16	3	0.0029	0.0419	0.0032
15	2	0.0033	0.0549	0.0056

Table A5.12 – unprocessed – log sigmoid

inputs	cells	mean	max	std
*17	4	0.0026	0.0388	0.0033
16	5	0.0028	0.0392	0.0035
*17	2	0.0028	0.0392	0.0032
17	3	0.0026	0.0394	0.0035
16	3	0.0025	0.0396	0.0033
14	3	0.0028	0.0396	0.0032
15	4	0.0028	0.0398	0.0034
14	5	0.0026	0.04	0.0035
15	3	0.0027	0.0404	0.0032
14	2	0.0027	0.0406	0.0032
16	2	0.0027	0.0407	0.0032
14	4	0.0027	0.0408	0.0031
15	2	0.0027	0.0408	0.0032
17	5	0.0028	0.0409	0.0033
16	4	0.0028	0.0416	0.0032
15	5	0.0059	0.3563	0.0218

Table A5.13 – unprocessed – tanh sigmoid

inputs	cells	mean	max	std
*6	2	0.0102	0.0328	0.0072
17	4	0.0027	0.0382	0.0031
16	5	0.0027	0.0384	0.004
15	4	0.0028	0.0384	0.0036
14	2	0.0028	0.0385	0.0037
17	5	0.0027	0.0388	0.0032
17	2	0.0027	0.0391	0.0034
14	3	0.0026	0.0392	0.0039
16	3	0.0027	0.0398	0.0033
15	5	0.0045	0.04	0.0041
14	5	0.0032	0.0402	0.0034
15	2	0.0028	0.0405	0.0035
16	4	0.0028	0.0405	0.0032
14	4	0.0026	0.0406	0.0036
15	3	0.0028	0.041	0.0032
17	3	0.0068	0.0451	0.0056

Table A5.14 – '÷100' – log sigmoid

inputs	cells	mean	max	std
*5	3	0.0028	0.0383	0.0036
16	3	0.0027	0.0385	0.0032
16	4	0.0035	0.0386	0.0034
17	4	0.0025	0.0388	0.0036
14	5	0.0028	0.0394	0.0032
17	5	0.0026	0.0395	0.0036
*6	2	0.0028	0.0395	0.0035
15	5	0.0025	0.0397	0.0037
17	3	0.0027	0.0397	0.0033
14	2	0.0026	0.0399	0.0032
14	4	0.0027	0.04	0.0033
15	2	0.0027	0.0403	0.0031
14	3	0.0027	0.0405	0.0034
15	4	0.0025	0.0409	0.0037
17	2	0.0028	0.041	0.0032
16	5	0.005	0.0475	0.0051

Table A5.15 – '÷100' – tanh sigmoid

inputs	*cells*	*mean*	*max*	*std*
16	4	0.0016	0.0195	0.0016
14	4	0.0013	0.0196	0.0017
15	3	0.0012	0.0199	0.0015
14	2	0.0012	0.02	0.0015
14	5	0.0016	0.0201	0.0018
17	2	0.0016	0.0202	0.0019
16	5	0.0013	0.0203	0.0017
16	2	0.0016	0.0203	0.0019
14	3	0.0017	0.0203	0.0018
15	4	0.0016	0.0204	0.0019
15	2	0.0012	0.0205	0.0016
17	5	0.0015	0.0207	0.0017
17	3	0.0016	0.0207	0.0019
17	4	0.0017	0.0207	0.0018
16	3	0.0017	0.0212	0.0018
15	5	0.0016	0.0215	0.0017

Table A5.16 – '*m* = 2' – log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*16	4	0.0038	0.0415	0.004
14	4	0.0035	0.0416	0.004
15	3	0.0034	0.0418	0.0038
14	2	0.0035	0.0419	0.0039
14	5	0.0039	0.0421	0.0042
16	5	0.0035	0.0422	0.0039
14	3	0.0041	0.0422	0.0044
*16	2	0.0041	0.0422	0.0045
17	2	0.0041	0.0422	0.0045
15	4	0.0041	0.0423	0.0044
15	2	0.0034	0.0424	0.0039
17	4	0.0041	0.0426	0.0044
17	5	0.0037	0.0427	0.004
17	3	0.0041	0.0427	0.0044
16	3	0.0041	0.0432	0.0044
15	5	0.0039	0.0434	0.0043

Table A5.16a – post-processed

inputs	*cells*	*mean*	*max*	*std*
17	3	0.0012	0.0189	0.0014
15	5	0.0016	0.0196	0.0017
17	5	0.0015	0.02	0.0018
15	2	0.0017	0.0202	0.0018
16	3	0.0015	0.0203	0.0016
16	5	0.0015	0.0203	0.0018
17	2	0.0015	0.0203	0.0018
16	2	0.0014	0.0204	0.0016
14	5	0.0016	0.0205	0.0019
17	4	0.0014	0.0206	0.0018
14	4	0.0014	0.0207	0.0017
16	4	0.0016	0.0209	0.0017
15	4	0.0016	0.0212	0.0018
15	3	0.0014	0.0378	0.0028
14	2	0.0016	0.0454	0.0027
14	3	0.0015	0.0482	0.0026

Table A5.17 – '*m* = 2' tanh sigmoid

inputs	*cells*	*mean*	*max*	*std*
*17	3	0.0034	0.0408	0.0038
15	5	0.0038	0.0416	0.004
17	5	0.0037	0.042	0.0039
16	5	0.0038	0.0422	0.0041
*14	2	0.0039	0.0422	0.0045
15	2	0.0041	0.0422	0.0044
16	2	0.0036	0.0423	0.0039
16	3	0.0037	0.0423	0.004
17	2	0.0039	0.0423	0.0043
15	3	0.0037	0.0424	0.0045
17	4	0.0037	0.0425	0.0041
14	5	0.004	0.0425	0.0044
14	4	0.0037	0.0426	0.004
16	4	0.004	0.0428	0.0043
15	4	0.0039	0.0432	0.0043
14	3	0.0037	0.0494	0.0044

Table A5.17a – post-processed

p.p.	*function*	*max error range (%)*	
		low input networks	*high input networks*
none	log	3.29 – 4.31	3.33 – 5.49
	tanh	3.91 – 3.99	3.88 – 35.63
÷100	log	3.59 – 3.99	3.28 – 4.51
	tanh	3.62 – 6.70	3.83 – 4.75
m = 2	log	1.95 – 6.79 (4.14 – 6.90)	1.95 – 2.15 (4.15 – 4.34)
	tanh	1.96 – 16.49 (4.15 – 16.62)	1.89 – 4.82 (4.08 – 4.94)

Table A5.18

Table A5.18 summarises the above results in terms of maximum error ranges, while Table A5.19 shows potential network solutions.

p/p	struct.	Function	mean	max	s.d.
none	5+2+K	Log	1.24	3.29	0.89
	17+3+K		1.24	3.30	0.91
	14+2+K		0.99	3.74	0.72
	4+2+K	Tanh	0.29	3.91	0.34
	17+4+K		0.26	3.88	0.33
	17+2+K		0.28	3.92	0.32
÷100	4+6+K	log	0.29	3.59	0.33
	6+2+K		0.28	3.75	0.37
	16+2+K		1.02	3.28	0.72
	4+4+K	tanh	0.29	3.62	0.37
	3+2+K		0.29	3.89	0.37
	15+3+K		0.28	3.83	0.36
	16+2+K		0.28	3.95	0.35
$m = 2$	4+4+K	log	**0.34	**4.14	**0.39
	3+2+K		**0.36	**4.20	**0.41
	16+4+K		**0.38	**4.15	**0.40
	14+2+K		**0.35	**4.19	**0.39
	5+5+K	tanh	**0.33	**4.15	**0.37
	5+2+K		**0.35	**4.20	**0.44
	17+3+K		**0.34	**4.08	**0.38
	14+2+K		**0.39	**4.22	**0.45

**after post-processing

Table A5.19

Examination of Table A5.18 shows that the tanh sigmoid based solutions with the higher number of inputs exhibit lower maximum error ranges than those with fewer inputs. Whilst the '$m = 2$' networks are consistently above 4% for maximum error values, in three cases the upper limit is lower than those of other network types, although the post-processed result for the tanh solution with few inputs has an extremely high upper limit of 16.62%

Table A5.19 suggests a relationship between maximum error, mean error and standard deviation; as the former rises the other two parameter values fall. The solution with the least maximum error was achieved by a log sigmoid network with 16 inputs and 2 hidden cells. However the second highest accuracy was achieved by a 5 input (2 hidden cells) log sigmoid network using data without pre-processing, and which would therefore be less demanding in terms of computational requirements.

A5.1.3 Predicting two intervals ahead (linear networks)

Inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0053	0.0412	0.0053	0
*4	0.0053	0.0409	0.0052	0
5	0.0052	0.0415	0.0049	0
6	0.0049	0.0428	0.0046	0
7	0.0047	0.0435	0.0047	0
8	0.0046	0.0431	0.0046	0
9	0.0046	0.0431	0.0046	1
10	0.0046	0.0434	0.0046	0
11	0.0046	0.0435	0.0046	0
12	0.0046	0.0432	0.0046	1
13	0.0046	0.0427	0.0047	0
14	0.0046	0.0428	0.0047	0
15	0.0046	0.0427	0.0047	0
16	0.0046	0.0427	0.0047	0
17	0.0046	0.0427	0.0047	0
18	0.0046	0.0427	0.0047	1
19	0.0046	0.0431	0.0047	0
20	0.0046	0.0432	0.0047	0

Table A5.20 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0053	0.0412	0.0053	0
*4	0.0053	0.041	0.0053	0
5	0.0051	0.0416	0.0049	0
6	0.0049	0.043	0.0046	0
7	0.0047	0.0435	0.0047	0
8	0.0046	0.0432	0.0046	1
9	0.0046	0.0431	0.0046	1
10	0.0046	0.0434	0.0046	0
11	0.0046	0.0435	0.0046	0
12	0.0046	0.0432	0.0046	1
13	0.0046	0.0427	0.0046	1
14	0.0045	0.0429	0.0047	2
15	0.0046	0.0427	0.0047	0
16	0.0046	0.0427	0.0047	0
17	0.0046	0.0427	0.0047	0
18	0.0046	0.0427	0.0047	2
19	0.0046	0.0429	0.0047	0
20	0.0046	0.0432	0.0047	0

Table A5.21 – '÷100'

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0016	0.0208	0.0018	0
*4	0.0015	0.0207	0.0016	0
5	0.0015	0.0209	0.0016	0
6	0.0014	0.021	0.0016	0
7	0.0014	0.021	0.0016	0
8	0.0014	0.0211	0.0016	0
9	0.0014	0.0212	0.0016	1
10	0.0014	0.0212	0.0016	0
11	0.0014	0.0211	0.0016	1
12	0.0014	0.021	0.0016	0
13	0.0014	0.0209	0.0016	0
14	0.0014	0.0209	0.0016	1
15	0.0014	0.0209	0.0016	2
16	0.0014	0.0209	0.0016	2
17	0.0014	0.0209	0.0016	1
18	0.0014	0.021	0.0016	0
19	0.0014	0.021	0.0016	2
20	0.0014	0.0209	0.0016	2

Table A5.22 – '*m* = 2'

inputs	*mean*	*max*	*s.d.*
3	0.0059	0.046	0.0061
*4	0.0057	0.046	0.006
5	0.0056	0.0462	0.0058
6	0.0056	0.0463	0.0057
7	0.0056	0.0462	0.0057
8	0.0056	0.0464	0.0057
9	0.0056	0.0464	0.0057
10	0.0056	0.0465	0.0057
11	0.0056	0.0464	0.0057
12	0.0056	0.0462	0.0057
13	0.0056	0.0461	0.0057
14	0.0056	0.0461	0.0057
15	0.0056	0.0462	0.0057
16	0.0056	0.0461	0.0057
17	0.0056	0.0461	0.0058
18	0.0056	0.0463	0.0058
19	0.0056	0.0462	0.0058
20	0.0057	0.0462	0.0058

Table A5.22a – post processed

The maximum error ranges for each pre-processing method are shown in 7.23.

p.p.	*max error range (%)*
none	4.09 – 4.35
÷100	4.10 – 4.35
m = 2	2.07 – 2.12
	4.60 – 4.65**

** after post-processing

Table A5.23

Solutions which demonstrate low maximum error are indicated above with an asterisk and are summarised below in Table A5.24 (errors are expressed as percentages) :

p.p	*struct*	*Mean*	*max*	*s.d.*	*<0.01*
none	4+K	0.53	4.09	0.53	0
÷ 100	4+K	0.53	4.10	0.53	0
m = 2	4+K	0.15	2.07	0.16	0
m = 2**	3+K	0.57	4.60	0.60	0

**after post-processing

Table A5.24

All solutions have maximum error ranges greater than 4%; the network trained without pre-processed data offers the most accurate solution whilst the '*m* = 2' network is the least accurate.

A5.1.4 Predicting two intervals ahead (non-linear networks)

inputs	*cells*	*mean*	*max*	*std*
*4	2	0.0124	0.0329	0.0089
6	2	0.0124	0.033	0.0089
3	5	0.0101	0.0331	0.0071
4	5	0.0052	0.0352	0.0053
5	3	0.0047	0.0383	0.0047
4	4	0.005	0.0395	0.0052
3	3	0.0051	0.0395	0.0054
6	4	0.0042	0.0398	0.0046
3	4	0.0054	0.0402	0.0052
3	2	0.0054	0.0404	0.0053
6	5	0.0042	0.0406	0.0046
5	2	0.0052	0.0406	0.0049
5	4	0.0046	0.0407	0.0049
6	3	0.0049	0.0422	0.0046
5	5	0.0047	0.0495	0.005
4	3	0.0056	0.1914	0.0096

Table A5.25 – unprocessed log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*5	2	0.0048	0.0359	0.0048
4	4	0.0052	0.0372	0.0051
3	3	0.005	0.0376	0.0051
5	5	0.0049	0.039	0.0047
5	3	0.0045	0.0393	0.0046
3	4	0.0051	0.0395	0.0052
3	5	0.0051	0.0395	0.0053
6	5	0.0043	0.0399	0.0043
5	4	0.0053	0.0399	0.0049
4	3	0.0054	0.0401	0.0052
3	2	0.0054	0.0404	0.0053
6	4	0.0043	0.0409	0.0049
6	2	0.0049	0.0417	0.0046
6	3	0.0049	0.0417	0.0046
4	5	0.0052	0.0574	0.0056
4	2	0.0107	0.0579	0.0079

Table A5.26 – unprocessed tanh sigmoid

Inputs	*cells*	*mean*	*max*	*std*
*5	2	0.0048	0.0356	0.0048
3	3	0.0058	0.0359	0.0053
4	3	0.0053	0.0362	0.0054
5	3	0.0045	0.0367	0.0045
5	5	0.0048	0.0367	0.005
3	5	0.005	0.0369	0.0051
4	4	0.0051	0.0375	0.0051
3	2	0.0052	0.0375	0.0053
3	4	0.0051	0.0381	0.0051
6	5	0.0044	0.0394	0.0046
6	3	0.0042	0.0398	0.0046
4	2	0.0057	0.0399	0.0052
6	2	0.0043	0.0411	0.0049
6	4	0.0047	0.0678	0.0057
4	5	0.0053	0.0803	0.0061
5	4	0.005	0.0825	0.006

Table A5.27 – '÷100' log sigmoid

inputs	*cells*	*mean*	*Max*	*std*
*5	2	0.0048	0.0357	0.0048
4	3	0.0053	0.0361	0.0054
4	5	0.0059	0.0366	0.0054
3	4	0.0053	0.0367	0.005
3	3	0.0055	0.0375	0.0051
3	5	0.0052	0.038	0.0053
3	2	0.005	0.0386	0.0054
4	4	0.0056	0.039	0.0052
6	5	0.0042	0.0396	0.0046
5	5	0.0055	0.0396	0.0051
6	4	0.005	0.0426	0.0049
4	2	0.0094	0.051	0.0073
5	4	0.0047	0.061	0.0056
5	3	0.0075	0.0655	0.0071
6	2	0.0075	0.0697	0.0062
6	3	0.0049	0.1183	0.0078

Table A5.28 – '÷100' tanh sigmoid

Inputs	*cells*	*mean*	*Max*	*std*
4	4	0.0013	0.0195	0.0015
3	5	0.0014	0.0197	0.0016
4	5	0.0013	0.0202	0.0015
6	2	0.0016	0.0202	0.0019
6	5	0.0012	0.0203	0.0015
6	3	0.0013	0.0203	0.0015
5	2	0.0016	0.0203	0.0018
5	3	0.0013	0.0204	0.0015
5	5	0.0013	0.0205	0.0015
3	2	0.0016	0.0207	0.0018
4	2	0.0015	0.0236	0.002
5	4	0.0016	0.0251	0.002
3	3	0.0016	0.027	0.0021
3	4	0.0017	0.0625	0.0033
6	4	0.0017	0.0662	0.0034
4	3	0.0025	0.3632	0.018

Table A5.29 – '*m* = 2' log sigmoid

inputs	*cells*	*mean*	*Max*	*std*
*4	4	0.0055	0.0447	0.0057
3	5	0.0057	0.0449	0.0059
4	5	0.0055	0.0454	0.0056
*6	2	0.0057	0.0454	0.0059
5	3	0.0055	0.0456	0.0056
6	3	0.0055	0.0456	0.0056
6	5	0.0055	0.0456	0.0056
4	2	0.0058	0.0456	0.0061
5	2	0.0058	0.0456	0.0059
5	5	0.0055	0.0457	0.0056
3	2	0.006	0.046	0.0061
5	4	0.0059	0.0464	0.0061
3	3	0.0059	0.0483	0.0062
3	4	0.0059	0.0845	0.0068
6	4	0.0057	0.0883	0.0069
4	3	0.0067	0.3914	0.0197

Table A5.29a – post-processed

inputs	*cells*	*mean*	*max*	*std*
5	3	0.0016	0.0198	0.0018
6	4	0.0013	0.0199	0.0015
4	2	0.0016	0.0201	0.0017
4	5	0.0015	0.0203	0.0018
6	3	0.0013	0.0205	0.0015
5	2	0.0014	0.0206	0.0016
3	4	0.0016	0.0206	0.0017
4	3	0.0016	0.0206	0.0017
3	5	0.0015	0.0208	0.0018
6	2	0.0015	0.0225	0.0019
6	5	0.0015	0.0258	0.0019
5	5	0.0014	0.0294	0.0019
3	2	0.0016	0.0491	0.0027
4	4	0.0016	0.0761	0.0036
5	4	0.0016	0.0833	0.0047
3	3	0.0019	0.154	0.0069

Table A5.30 – '*m* = 2' tanh sigmoid

inputs	*cells*	*mean*	*max*	*std*
*6	2	0.0054	0.045	0.0056
5	3	0.0055	0.045	0.0057
6	4	0.0055	0.0451	0.0056
4	2	0.0057	0.0453	0.006
5	5	0.0055	0.0455	0.0057
4	5	0.0057	0.0456	0.006
6	3	0.0055	0.0457	0.0056
3	4	0.0059	0.0458	0.006
5	2	0.0056	0.0459	0.0058
4	3	0.0058	0.0459	0.006
3	5	0.0059	0.046	0.006
3	2	0.006	0.0461	0.0062
6	5	0.0056	0.047	0.0059
4	4	0.0058	0.0713	0.0066
5	4	0.0058	0.0852	0.007
3	3	0.0062	0.16	0.0091

Table A5.30a – post-processed

p.p.	*function*	*max error range (%)*
none	log	3.29 – 19.14
	tanh	3.59 – 5.79
÷100	log	3.56 – 8.25
	tanh	3.57 – 11.83
m = 2	log	1.95 – 36.32 (4.47 – 39.14)
	tanh	1.98 – 15.40 (4.50 – 16.00)

Table A5.31

Table A5.31 summarises the above results in terms of maximum error ranges, while table A5.32 shows potential solutions.

pre-proc?	*struct.*	*function*	*Mean*	*max*	*s.d.*
none	4+2+K	log	1.24	3.29	0.89
	5+2+K	tanh	0.48	3.59	0.48
÷100	5+2+K	log	0.48	3.56	0.48
	5+5+K	tanh	0.48	3.57	0.48
m = 2	4+4+K	log	**0.55	**4.47	**0.59
	6+2+K		**0.57	**4.54	**0.59
	6+2+K	tanh	**0.54	**4.50	**0.56

**after post-processing

Table A5.32

The upper limits of the maximum error ranges are greatly increased over those of the corresponding linear solutions and over those of the one sampling interval ahead results. A log sigmoid network trained without pre-processing offers the most accurate solution

although the mean error and standard deviation are nearly twice those of the other networks. Again, as the maximum error is reduced there is a corresponding increase in mean error and standard deviation.

A5.2 Investigating CV Prediction with Original 23-sec data

A5.2.1 One interval ahead (linear networks)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.001	0.0434	0.0017	0
4	0.0009	0.0436	0.0017	0
5	0.001	0.0439	0.0017	0
6	0.0009	0.0439	0.0017	0
7	0.0009	0.0439	0.0017	0
8	0.0009	0.0438	0.0017	0
9	0.0009	0.0438	0.0017	0
10	0.0009	0.0439	0.0017	0
11	0.001	0.044	0.0017	1
12	0.001	0.044	0.0017	1
13	0.0009	0.044	0.0016	1
14	0.001	0.044	0.0016	1
15	0.001	0.044	0.0017	3
16	0.001	0.044	0.0017	4
17	0.001	0.0439	0.0016	1
18	0.0009	0.0441	0.0016	1
19	0.0009	0.0442	0.0016	1
20	0.0009	0.0442	0.0016	2

Table A5.33 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.001	0.0434	0.0017	0
4	0.0009	0.0436	0.0017	0
5	0.001	0.0439	0.0017	0
6	0.0009	0.0439	0.0017	0
7	0.0009	0.0439	0.0017	0
8	0.0009	0.0438	0.0017	0
9	0.0009	0.0438	0.0017	0
10	0.0009	0.0439	0.0017	0
11	0.001	0.044	0.0017	1
12	0.001	0.044	0.0017	1
13	0.0009	0.044	0.0016	2
14	0.001	0.044	0.0016	1
15	0.001	0.044	0.0017	3
16	0.001	0.044	0.0017	4
17	0.001	0.0439	0.0016	1
18	0.0009	0.0441	0.0016	1
19	0.0009	0.0441	0.0016	1
20	0.0009	0.0442	0.0016	2

Table A5.34 – '÷100'

inputs	mean	max	s.d.	<0.01
3	0.0005	0.0219	0.0009	0
4	0.0005	0.0221	0.0008	0
5	0.0005	0.0221	0.0009	0
6	0.0005	0.0221	0.0008	0
7	0.0005	0.022	0.0008	0
8	0.0005	0.022	0.0008	0
9	0.0005	0.022	0.0008	2
10	0.0005	0.0221	0.0008	1
11	0.0005	0.0221	0.0008	2
12	0.0005	0.0221	0.0008	1
13	0.0005	0.0221	0.0008	1
14	0.0005	0.022	0.0008	2
15	0.0005	0.022	0.0008	2
16	0.0005	0.022	0.0008	1
17	0.0005	0.0221	0.0008	1
18	0.0005	0.0221	0.0008	1
19	0.0005	0.0221	0.0008	2
20	0.0005	0.0221	0.0008	3

Table A5.35 – '*m* = 2'

inputs	mean	max	s.d.
3	0.0016	0.0427	0.0019
4	0.0016	0.0429	0.0019
5	0.0016	0.0429	0.0019
6	0.0016	0.0429	0.0019
7	0.0016	0.0428	0.0019
8	0.0016	0.0428	0.0019
9	0.0016	0.0429	0.0019
10	0.0016	0.0429	0.0019
11	0.0016	0.0429	0.0019
12	0.0016	0.0429	0.0019
13	0.0016	0.0429	0.0019
14	0.0016	0.0429	0.0019
15	0.0016	0.0429	0.0019
16	0.0016	0.0428	0.0019
17	0.0015	0.0429	0.0019
18	0.0015	0.0429	0.0019
19	0.0015	0.0429	0.0019
20	0.0015	0.0429	0.0019

Table A5.35a – post-processed

The maximum error ranges for each pre-processing method are shown in A5.36.

p-p	max error range (%)
none	4.34 – 4.42
÷100	4.34 – 4.42
m = 2	2.19 – 2.21
m = 2	4.27 – 4.29**

** after post-processing

Table A5.36

Solutions which demonstrate low maximum error are indicated with and asterisk and are summarised below in Table A5.37 (errors are expressed as percentages) :

p/p	struct	Mean	max	s.d.	<0.01
none	3+K	0.10	4.34	0.17	0
÷ 100	3+K	0.10	4.34	0.17	0
m = 2	3+K	0.05	2.19	0.09	0
m = 2**	3+K	0.16	4.27	0.19	0

**after post-processing

Table A5.37

A5.2.2 One interval ahead (non-linear)

inputs	*cells*	*mean*	*max*	*std*
3	2	0.001	0.0433	0.0017
4	4	0.0009	0.0434	0.0017
3	5	0.001	0.0434	0.0016
6	4	0.0009	0.0435	0.0016
3	3	0.001	0.0435	0.0017
3	4	0.001	0.0435	0.0017
4	2	0.001	0.0435	0.0017
6	3	0.0009	0.0436	0.0017
4	3	0.001	0.0436	0.0017
5	5	0.0009	0.0437	0.0017
6	2	0.0009	0.0438	0.0017
6	5	0.001	0.0438	0.0017
5	2	0.001	0.0439	0.0017
5	3	0.001	0.0439	0.0017
5	4	0.001	0.0439	0.0017
4	5	0.0009	0.0473	0.002

Table A5.38 – log sigmoid unprocessed

inputs	*cells*	*mean*	*max*	*std*
3	2	0.0122	0.0327	0.0087
3	3	0.001	0.0433	0.0017
3	5	0.001	0.0433	0.0017
3	4	0.001	0.0434	0.0016
4	5	0.0009	0.0436	0.0019
6	4	0.0009	0.0436	0.0017
4	2	0.001	0.0436	0.0017
4	4	0.001	0.0436	0.0017
4	3	0.0009	0.0438	0.0016
5	5	0.001	0.0438	0.0019
6	2	0.001	0.0438	0.0017
5	2	0.001	0.0439	0.0017
5	3	0.001	0.0439	0.0017
5	4	0.001	0.044	0.0017
6	5	0.001	0.044	0.002
6	3	0.0009	0.0441	0.0017

Table A5.39 – tanh sigmoid unprocessed

inputs	*cells*	*mean*	*max*	*std*
6	2	0.0038	0.038	0.0039
5	2	0.003	0.0413	0.0028
3	4	0.0011	0.0429	0.0016
4	3	0.001	0.043	0.0017
3	3	0.0013	0.0431	0.0017
4	4	0.0009	0.0432	0.0016
5	5	0.001	0.0433	0.0016
3	2	0.001	0.0434	0.0016
4	2	0.001	0.0434	0.0017
6	4	0.0009	0.0435	0.0016
6	5	0.001	0.0435	0.0016
4	5	0.0009	0.0436	0.0018
n6	3	0.0009	0.0436	0.0017
5	3	0.0009	0.0437	0.0017
3	5	0.0014	0.0437	0.0017
5	4	0.0009	0.0438	0.0016

Table A5.40 – '÷100' log sigmoid

inputs	*cells*	*mean*	*max*	*std*
6	2	0.0034	0.0375	0.003
5	4	0.0015	0.0406	0.0017
6	4	0.0019	0.0424	0.0023
3	4	0.001	0.0429	0.0017
6	5	0.001	0.0431	0.0017
3	5	0.001	0.0432	0.0017
3	2	0.001	0.0433	0.0017
5	5	0.0009	0.0434	0.0015
6	3	0.0009	0.0434	0.0017
3	3	0.001	0.0434	0.0016
4	2	0.001	0.0434	0.0017
4	3	0.001	0.0436	0.0016
4	4	0.001	0.0436	0.0018
4	5	0.0009	0.0437	0.0016
5	2	0.001	0.0439	0.0017
5	3	0.0037	0.0909	0.0044

Table A5.41 – '÷100' tanh sigmoid

inputs	cells	mean	max	std
3	5	0.0004	0.0219	0.0007
4	2	0.0004	0.0219	0.0007
4	4	0.0004	0.022	0.0007
4	5	0.0004	0.022	0.0007
6	2	0.0004	0.022	0.0008
6	4	0.0005	0.022	0.0008
5	2	0.0004	0.0221	0.0008
5	3	0.0004	0.0221	0.0007
6	3	0.0004	0.0221	0.0007
5	4	0.0005	0.0221	0.0008
6	5	0.0005	0.0483	0.0016
5	5	0.0005	0.0642	0.0025
4	3	0.001	0.6186	0.0169
3	2	0.0011	0.657	0.0178
3	3	0.0011	0.85	0.023
3	4	0.0059	7.5182	0.2034

Table A5.42 – '*m* = 2' log sigmoid

inputs	cells	mean	max	std
4	2	0.0015	0.0427	0.002
3	5	0.0015	0.0428	0.002
4	4	0.0015	0.0428	0.002
6	2	0.0015	0.0428	0.0019
4	5	0.0015	0.0429	0.0019
5	2	0.0015	0.0429	0.0019
5	3	0.0015	0.0429	0.002
6	3	0.0015	0.0429	0.0019
5	4	0.0016	0.0429	0.0021
6	4	0.0016	0.0429	0.002
6	5	0.0017	0.05	0.0025
5	5	0.0016	0.065	0.003
4	3	0.002	0.6186	0.017
3	2	0.0023	0.6568	0.0179
3	3	0.0022	0.8489	0.023
3	4	0.007	7.5172	0.2035

Table R42a – post-processed

Inputs	cells	mean	max	std
3	3	0.0004	0.0219	0.0007
5	2	0.0004	0.0219	0.0008
6	4	0.0004	0.0219	0.0007
3	2	0.0005	0.0219	0.0008
3	5	0.0004	0.022	0.0007
5	5	0.0004	0.022	0.0008
4	3	0.0004	0.0221	0.0007
5	3	0.0004	0.0221	0.0007
5	4	0.0005	0.0221	0.0008
6	3	0.0005	0.0221	0.0008
4	5	0.0004	0.0306	0.0012
6	5	0.0005	0.0331	0.0013
4	4	0.0006	0.1926	0.0054
6	2	0.0012	0.4989	0.0152
3	4	0.0008	0.57	0.0154
4	2	0.0012	0.6749	0.0183

Table A5.43 – '*m* = 2' tanh sigmoid

inputs	cells	mean	max	std
3	3	0.0015	0.0427	0.002
5	2	0.0015	0.0427	0.002
6	4	0.0015	0.0427	0.002
3	2	0.0016	0.0427	0.002
3	5	0.0015	0.0428	0.002
4	5	0.0015	0.0428	0.002
5	5	0.0015	0.0429	0.0019
5	4	0.0016	0.0429	0.002
6	3	0.0016	0.0429	0.002
6	5	0.0016	0.0429	0.0024
4	3	0.0015	0.043	0.002
5	3	0.0015	0.043	0.002
4	4	0.0017	0.1931	0.0057
6	2	0.0023	0.499	0.0151
3	4	0.0019	0.57	0.0155
4	2	0.0023	0.6739	0.0184

Table A5.43a – post-processed

p.p.	*function*	*max error range (%)*
none	log	4.33 – 4.73
	tanh	3.27 – 4.41
÷100	log	3.80 – 4.38
	tanh	3.75 – 9.09
m = 2	log	4.27 – 751.72
	tanh	4.27 – 67.39

Table A5.44

Table A5.44 summarises the above results in terms of maximum error ranges, while table A5.45 shows potential network solutions.

p/p	*struct.*	*function*	*mean*	*max*	*s.d.*
none	3+2+K	log	0.10	4.33	0.17
	3+2+K	tanh	1.22	3.27	0.87
÷100	6+2+K	log	0.38	3.80	0.39
	6+2+K	tanh	0.34	3.75	0.30
m = 2**	4+2+K	log	0.15	4.27	0.20
	3+3+K	tanh	0.15	4.27	0.20
	5+2+K		0.15	4.27	0.20

**after post-processing

Table A5.45

A5.2.3 Two intervals ahead (linear)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.002	0.05	0.0028	0
4	0.002	0.0509	0.0028	0
5	0.002	0.049	0.0028	0
6	0.002	0.0509	0.0029	0
7	0.0019	0.05	0.0029	0
8	0.0019	0.0495	0.0028	0
9	0.0019	0.0495	0.0028	0
10	0.002	0.05	0.0029	0
11	0.002	0.0485	0.0028	0
12	0.002	0.046	0.0027	0
13	0.0019	0.045	0.0027	0
14	0.0019	0.044	0.0027	0
15	0.0019	0.0443	0.0027	0
*16	0.0019	0.0438	0.0027	0
17	0.0019	0.0442	0.0027	0
18	0.0019	0.0443	0.0026	0
19	0.0019	0.0443	0.0026	0
20	0.0019	0.0443	0.0026	0

Table A5.46 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.002	0.0499	0.0028	0
4	0.002	0.0508	0.0028	0
5	0.002	0.0491	0.0028	0
6	0.002	0.0511	0.0029	0
7	0.0019	0.05	0.0029	0
8	0.0019	0.0495	0.0028	0
9	0.0019	0.0496	0.0028	0
10	0.002	0.0499	0.0028	0
11	0.002	0.0485	0.0028	0
12	0.002	0.0459	0.0027	0
13	0.0019	0.0449	0.0027	0
14	0.0019	0.0441	0.0027	0
15	0.0019	0.0443	0.0027	0
*16	0.0019	0.0438	0.0027	0
17	0.0019	0.0441	0.0027	0
18	0.0019	0.0443	0.0026	0
19	0.0019	0.0443	0.0026	0
20	0.0019	0.0443	0.0026	0

Table A5.47 – '÷100'

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0006	0.0221	0.0009	0
4	0.0006	0.0221	0.0009	1
5	0.0006	0.0221	0.0009	0
6	0.0006	0.022	0.0009	0
7	0.0006	0.022	0.0009	0
8	0.0006	0.022	0.0009	0
9	0.0006	0.0221	0.0009	0
10	0.0006	0.0221	0.0009	1
11	0.0006	0.0221	0.0009	0
12	0.0006	0.0221	0.0009	0
13	0.0006	0.022	0.0009	1
14	0.0006	0.0221	0.0009	0
15	0.0006	0.022	0.0009	0
16	0.0006	0.0221	0.0008	0
17	0.0006	0.0222	0.0008	0
18	0.0006	0.0222	0.0008	0
19	0.0006	0.0222	0.0008	0
20	0.0006	0.0222	0.0008	1

Table A5.48 – '*m* = 2'

inputs	*mean*	*max*	*s.d.*
*3	0.0017	0.0425	0.0021
4	0.0017	0.0425	0.0021
5	0.0017	0.0426	0.0021
6	0.0017	0.0425	0.0021
7	0.0017	0.0425	0.0021
8	0.0017	0.0425	0.0021
9	0.0017	0.0425	0.0021
10	0.0017	0.0426	0.002
11	0.0017	0.0426	0.002
12	0.0017	0.0426	0.002
13	0.0017	0.0426	0.002
14	0.0017	0.0426	0.002
15	0.0017	0.0426	0.002
16	0.0017	0.0426	0.002
17	0.0017	0.0427	0.002
18	0.0017	0.0427	0.002
19	0.0017	0.0427	0.002
20	0.0017	0.0427	0.002

Table A5.48a – post-processed

The maximum error ranges for each pre-processing method are shown in table A5.49.

p-p	*max error range (%)*
none	4.38 – 5.09
÷100	4.38 – 5.11
m = 2	2.21 – 2.22
m = 2	4.25 – 4.27**

** after post-processing

Table A5.49

Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in table A5.50 (errors are expressed as percentages) :

p/p	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	16+K	0.19	4.38	0.27	0
÷ 100	16+K	0.19	4.38	0.27	0
m = 2	3+K	0.06	2.21	0.09	0
m = 2**	3+K	0.17	4.25	0.21	0

**after post-processing

Table A5.50

A5.2.4 Two intervals ahead (non-linear)

The above tables indicate some 15 - 17 inputs would be appropriate for this investigation with the exception of the '$m = 2$' method where some 3 – 5 inputs are indicated.

inputs	*cells*	*mean*	*max*	*std*
*15	3	0.0017	0.0424	0.0024
16	3	0.0018	0.0425	0.0026
16	5	0.0019	0.043	0.003
15	5	0.0017	0.0433	0.0027
*17	2	0.0017	0.0433	0.0027
16	2	0.0019	0.0436	0.0027
17	4	0.0017	0.0437	0.0025
15	2	0.0019	0.0437	0.0027
17	5	0.0019	0.0439	0.0026
17	3	0.0019	0.0441	0.0026
15	4	0.0017	0.0552	0.003
16	4	0.0031	0.4661	0.023

Table A5.51 – log sigmoid unprocessed

inputs	*cells*	*Mean*	*max*	*std*
*15	2	0.0101	0.0394	0.0074
15	4	0.0017	0.0427	0.0027
17	4	0.0017	0.0429	0.0027
17	5	0.0019	0.0429	0.0025
16	3	0.0019	0.043	0.003
15	3	0.0016	0.0433	0.0026
17	3	0.0017	0.0434	0.0027
16	4	0.0018	0.0439	0.0026
16	2	0.0019	0.0439	0.0026
17	2	0.0019	0.0446	0.0026
15	5	0.0091	0.0949	0.0068
16	5	0.003	0.6501	0.028

Table A5.52 – tanh sigmoid unprocessed

Inputs	*cells*	*mean*	*max*	*std*
*16	5	0.004	0.0389	0.0039
17	4	0.0018	0.0419	0.0028
17	5	0.0018	0.0419	0.0027
*17	3	0.0017	0.0426	0.0028
16	4	0.0017	0.0428	0.0028
15	3	0.0019	0.0434	0.0026
17	2	0.0019	0.0435	0.0026
16	2	0.0019	0.0436	0.0027
15	2	0.0019	0.0437	0.0028
15	4	0.0017	0.0438	0.0025
16	3	0.0051	0.0596	0.0052
15	5	0.0018	0.0857	0.0037

Table A5.53 – log sigmoid '÷100'

Inputs	*cells*	*Mean*	*max*	*std*
*17	4	0.0058	0.0402	0.0053
16	5	0.0024	0.0418	0.0029
17	5	0.0017	0.0422	0.0029
16	4	0.0019	0.0422	0.0027
*17	2	0.0017	0.0427	0.0027
16	2	0.0019	0.0431	0.0026
17	3	0.0017	0.0432	0.0028
16	3	0.0019	0.0433	0.0026
15	2	0.0019	0.0437	0.0027
15	4	0.0017	0.0438	0.0028
15	5	0.0024	0.0446	0.0032
15	3	0.002	0.0646	0.0035

Table A5.54 – tanh sigmoid '÷100'

inputs	cells	mean	max	std
5	2	0.0006	0.0218	0.0009
5	5	0.0005	0.022	0.0008
3	5	0.0006	0.0221	0.0008
4	2	0.0005	0.0222	0.0008
4	4	0.0005	0.0222	0.0008
3	3	0.0005	0.0223	0.0008
4	3	0.0005	0.0223	0.0008
4	5	0.0005	0.0226	0.0008
5	3	0.0005	0.0228	0.0008
3	2	0.0007	0.0569	0.0018
3	4	0.0009	0.2463	0.0091
5	4	0.0015	0.6806	0.0242

Table A5.55 – log sigmoid '*m* = 2'

inputs	cells	mean	max	std
*3	5	0.0027	0.0417	0.0032
5	5	0.0027	0.0417	0.0032
*5	2	0.0029	0.0417	0.0034
4	2	0.0027	0.0418	0.0032
4	4	0.0027	0.0418	0.0032
3	3	0.0027	0.0419	0.0032
4	3	0.0027	0.0419	0.0032
4	5	0.0027	0.0422	0.0032
5	3	0.0027	0.0425	0.0032
3	2	0.003	0.0561	0.0037
3	4	0.003	0.2469	0.0096
5	4	0.0037	0.674	0.0242

Table A5.55a – post-processed

inputs	cells	mean	max	std
5	5	0.0006	0.0221	0.0008
3	2	0.0005	0.0223	0.0008
4	4	0.0005	0.0224	0.0008
3	5	0.0006	0.0294	0.0011
4	5	0.0006	0.0318	0.0012
5	2	0.0008	0.0908	0.0032
4	2	0.0007	0.104	0.0029
5	3	0.0008	0.405	0.011
4	3	0.0011	0.6294	0.017
3	4	0.0013	0.7682	0.0217
3	3	0.0049	3.072	0.1043
5	4	0.0041	3.4241	0.0957

Table A5.56 – tanh sigmoid '*m* = 2'

inputs	cells	mean	max	std
*4	5	0.0027	0.0417	0.0033
5	5	0.0027	0.0417	0.0032
3	5	0.0027	0.0418	0.0033
*3	2	0.0027	0.042	0.0032
4	4	0.0027	0.042	0.0032
5	2	0.0031	0.0915	0.0046
4	2	0.0028	0.1047	0.0043
5	3	0.003	0.4039	0.0113
4	3	0.0032	0.6219	0.0171
3	4	0.0035	0.768	0.0219
3	3	0.0071	3.0746	0.1048
5	4	0.0063	3.4268	0.0959

Table A5.56a – post-processed

p.p.	function	max error range (%)
none	log	4.24 – 46.61
	tanh	3.94 – 65.01
÷100	log	3.89 – 8.57
	tanh	4.02 – 6.46
m = 2**	log	4.17 – 67.40
	tanh	4.17 – 342.68

** after post-processing

Table A5.57

Table A5.57 summarises the above results in terms of maximum error ranges, while table A5.58 (below) shows potential network solutions.

p/p	*struct.*	*function*	*mean*	*max*	*s.d.*
none	15+3+K	log	0.17	4.24	0.24
	17+2+K		0.17	4.33	0.27
	15+2+K	tanh	1.01	3.94	0.74
÷100	16+5+K	log	0.40	3.89	0.39
	17+3+K		0.17	4.26	0.28
	17+4+K	tanh	0.58	4.02	0.53
	17+2+K		0.17	4.27	0.27
m = 2**	3+5+K	log	0.27	4.17	0.32
	5+2+K		0.29	4.17	0.34
	4+5+K	tanh	0.27	4.17	0.33
	3+2+K		0.27	4.20	0.32

**after post-processing

Table A5.58

A5.3 Effect of possible mass spectrometer failure on CV results

Tables A5.59 – 7.60 show results for the data before the possible fault whilst tables A5.61 – 62 show those for the data after it.

inputs	*mean*	*max*	*s.d.*
3	0.0031	0.0174	0.003
4	0.0029	0.0176	0.003
*5	0.0029	0.0169	0.0028
6	0.0028	0.0196	0.0026
7	0.0028	0.0183	0.0026
8	0.0027	0.0186	0.0025
9	0.0027	0.0183	0.0025
10	0.0027	0.0184	0.0025
11	0.0027	0.0184	0.0025
12	0.0027	0.0187	0.0025
13	0.0027	0.0185	0.0025
14	0.0027	0.0184	0.0025
15	0.0027	0.0185	0.0025
16	0.0027	0.0184	0.0025
17	0.0027	0.0184	0.0025
18	0.0027	0.0182	0.0025
19	0.0027	0.0182	0.0025
20	0.0027	0.0182	0.0025

Table A5.59 – unprocessed 1-min

inputs	*mean*	*max*	*s.d.*
3	0.0010	0.0086	0.0010
4	0.0009	0.0085	0.0010
5	0.0009	0.0085	0.0010
*6	0.0009	0.0083	0.0010
7	0.0009	0.0084	0.0010
8	0.0009	0.0083	0.0010
9	0.0009	0.0083	0.0010
10	0.0009	0.0083	0.0010
11	0.0009	0.0085	0.0009
12	0.0009	0.0085	0.0009
13	0.0009	0.0086	0.0009
14	0.0009	0.0086	0.0009
15	0.0009	0.0086	0.0009
16	0.0009	0.0086	0.0009
17	0.0009	0.0087	0.0009
18	0.0009	0.0088	0.0009
19	0.0009	0.0088	0.0009
20	0.0009	0.0088	0.0009

Table A5.60 – unprocessed 23-sec

inputs	*mean*	*max*	*s.d.*
3	0.0031	0.0173	0.003
4	0.0029	0.0175	0.003
*5	0.0029	0.0169	0.0028
6	0.0028	0.0196	0.0026
7	0.0028	0.0180	0.0026
8	0.0027	0.0186	0.0025
9	0.0027	0.0183	0.0025
10	0.0027	0.0184	0.0025
11	0.0027	0.0184	0.0025
12	0.0027	0.0187	0.0025
13	0.0027	0.0185	0.0025
14	0.0027	0.0191	0.0025
15	0.0027	0.0194	0.0025
16	0.0027	0.0184	0.0025
17	0.0027	0.0184	0.0025
18	0.0027	0.0183	0.0025
19	0.0027	0.0182	0.0025
20	0.0027	0.0182	0.0025

Table A5.61 – '÷100' 1-min

inputs	*mean*	*max*	*s.d.*
3	0.0010	0.0087	0.001
4	0.0009	0.0085	0.001
5	0.0009	0.0085	0.001
*6	0.0009	0.0083	0.001
7	0.0009	0.0084	0.001
8	0.0009	0.0083	0.001
9	0.0009	0.0083	0.001
10	0.0009	0.0083	0.001
11	0.0009	0.0085	0.0009
12	0.0009	0.0085	0.0009
13	0.0009	0.0086	0.0009
14	0.0009	0.0086	0.0009
15	0.0009	0.0086	0.0009
16	0.0009	0.0086	0.0009
17	0.0009	0.0087	0.0009
18	0.0009	0.0088	0.0009
19	0.0009	0.0088	0.0009
20	0.0009	0.0088	0.0009

Table A5.62 – '÷100' 23-sec

inputs	mean	max	s.d.
3	0.0023	0.0125	0.0028
4	0.0023	0.0126	0.0026
5	0.0022	0.0114	0.0025
6	0.0022	0.0099	0.0022
7	0.0021	0.0104	0.0024
8	0.0020	0.0094	0.0022
9	0.0020	0.0093	0.0022
10	0.0020	0.0093	0.0022
11	0.0020	0.0092	0.0022
12	0.0020	0.0092	0.0022
13	0.0019	0.0090	0.0020
*14	0.0019	0.0091	0.0020
15	0.0019	0.0091	0.0020
16	0.0019	0.0092	0.0020
17	0.0019	0.0091	0.0020
18	0.0018	0.0091	0.0020
19	0.0018	0.0092	0.0020
20	0.0018	0.0091	0.0020

Table A5.63 –1-min

inputs	mean	max	s.d.
3	0.0009	0.0090	0.0010
4	0.0008	0.0091	0.0009
5	0.0008	0.0089	0.0009
6	0.0008	0.0090	0.0009
7	0.0008	0.0090	0.0009
8	0.0008	0.0090	0.0009
9	0.0008	0.0089	0.0009
10	0.0008	0.0090	0.0009
11	0.0008	0.0089	0.0009
12	0.0008	0.0087	0.0009
13	0.0008	0.0088	0.0008
14	0.0008	0.0088	0.0009
15	0.0008	0.0088	0.0009
16	0.0008	0.0088	0.0009
17	0.0008	0.0087	0.0009
18	0.0008	0.0087	0.0009
*19	0.0008	0.0086	0.0009
20	0.0008	0.0086	0.0009

Table A5.64 –23-sec

inputs	mean	max	s.d.
3	0.0023	0.0125	0.0028
4	0.0022	0.0126	0.0026
5	0.0022	0.0114	0.0025
6	0.0022	0.0099	0.0022
7	0.0021	0.0105	0.0024
8	0.0020	0.0094	0.0022
9	0.0020	0.0093	0.0022
10	0.0020	0.0093	0.0022
11	0.0020	0.0092	0.0022
12	0.0020	0.0092	0.0022
13	0.0019	0.0090	0.0020
14	0.0019	0.0095	0.0020
15	0.0018	0.0096	0.0019
16	0.0019	0.0092	0.0020
*17	0.0019	0.0091	0.0020
18	0.0018	0.0091	0.0020
19	0.0018	0.0092	0.0020
20	0.0018	0.0091	0.0020

Table A5.65 – '÷100' 1min

inputs	mean	max	s.d.
3	0.0009	0.0089	0.0010
4	0.0008	0.0091	0.0009
5	0.0008	0.0090	0.0009
6	0.0008	0.0090	0.0009
7	0.0008	0.0090	0.0009
8	0.0008	0.0090	0.0009
9	0.0008	0.0089	0.0009
10	0.0008	0.0090	0.0009
11	0.0008	0.0089	0.0009
12	0.0008	0.0087	0.0009
13	0.0008	0.0088	0.0008
14	0.0008	0.0088	0.0009
15	0.0008	0.0088	0.0009
16	0.0008	0.0088	0.0009
17	0.0008	0.0087	0.0009
18	0.0008	0.0087	0.0009
*19	0.0008	0.0086	0.0009
20	0.0008	0.0086	0.0009

Table A5.66 – '÷100' 23-sec

max error ranges				
pp	before		after	
	1-min	23-sec	1-min	23-sec
none	1.74 - 1.96	0.83 - 0.88	0.90 - 1.26	0.86 - 0.91
100	1.69 - 1.96	0.83 - 0.88	0.91 - 1.26	0.86 - 0.91

Table A5.67 – max percentage absolute error ranges

The above table shows that there is little justification for the extra computational effort require in pre-processing. Further, it indicates that the networks resume efficient operation after the disruption following the encounter with the suspected mass spectrometer fault.

		before				after			
pp	data	type	mean	max	std	type	mean	max	std
	1-min	5+K	0.29	1.69	0.28	14+k	0.19	0.91	0.20
	23-sec	6+K	0.09	0.83	0.10	19+k	0.08	0.86	0.09
÷100	1-min	5+K	0.29	1.69	0.28	17+K	0.19	0.91	0.2
	23-sec	6+k	0.09	0.83	0.10	19+K	0.08	0.86	0.09

Table A5.68 – suitable networks as indicated by the above results.

The later, shorter, data set suggests higher accuracy may be obtained from networks with a larger number of inputs, although this set would have less variation in dynamics than its larger counterpart. In both cases there is a gain in accuracy by utilising the 23-second data. With respect to the larger data set there is a 50% fall in maximum error, with the mean absolute error and the standard deviation reducing by some $^2/_3$. These results also confirm that pre-processing through division by 100 does not increase the efficiency of the network.

A5.4 Specific Gravity (s.g.)

The data lies within the overall range 0.28 – 0.32, suitable for direct input to a neural network; hence there was no requirement for pre-processing.

A5.4.1 Predicting 1 sampling interval ahead

Examination of Table A5.69 (results for the linear network investigation) shows that non-linear investigation should proceed with 12-14 input networks.

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0017	0.0107	0.0015	0
4	0.0017	0.0107	0.0015	0
5	0.0016	0.0107	0.0014	1
6	0.0016	0.0106	0.0014	1
7	0.0016	0.0106	0.0014	2
8	0.0016	0.0106	0.0014	1
9	0.0016	0.0106	0.0014	0
10	0.0016	0.0102	0.0014	1
11	0.0016	0.0098	0.0014	1
12	0.0016	0.0095	0.0014	0
*13	0.0016	0.0094	0.0014	2
14	0.0016	0.0096	0.0014	1
15	0.0016	0.0098	0.0014	0
16	0.0015	0.0096	0.0014	2
17	0.0015	0.0095	0.0014	3
18	0.0015	0.0094	0.0014	1
19	0.0015	0.0094	0.0014	2
20	0.0015	0.0094	0.0014	3

Table A5.69 – linear

inputs	*cells*	*mean*	*Max*	*std*
*13	3	0.0015	0.0084	0.0013
12	5	0.0014	0.0085	0.0012
14	3	0.0014	0.0086	0.0013
12	4	0.0015	0.0086	0.0013
12	3	0.0015	0.0088	0.0013
*12	2	0.0015	0.0092	0.0013
13	5	0.0015	0.0093	0.0013
13	2	0.0015	0.0095	0.0013
13	4	0.0016	0.0095	0.0014
14	4	0.0015	0.0096	0.0013
14	2	0.0015	0.0098	0.0013
14	5	0.0048	0.0256	0.004

Table A5.70 – log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*12	3	0.0014	0.0084	0.0012
13	5	0.0014	0.0086	0.0013
12	4	0.0015	0.0091	0.0013
12	5	0.0016	0.0091	0.0013
*12	2	0.0016	0.0092	0.0013
14	4	0.0015	0.0093	0.0013
14	2	0.0014	0.0096	0.0013
13	4	0.0015	0.0099	0.0013
14	5	0.0016	0.0124	0.0014
14	3	0.0014	0.0126	0.0013
13	2	0.0055	0.0216	0.0039
13	3	0.0106	0.0355	0.0078

Table A5.71 – tanh sigmoid

The maximum error ranges for each pre-processing method for the linear networks are shown in A5.72. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in Table A5.73 (errors are expressed as percentages) .

function	*max error range (%)*
linear	0.94 – 1.07
log	0.84 – 2.56
tanh	0.84 – 3.55

Table A5.72

function	*struct*	*mean*	*max*	*s.d.*	*<0.01*
linear	13+K	0.16	0.94	0.14	2
log	13+3K	0.15	0.84	0.13	—
	12+2K	0.15	0.92	0.13	—
tanh	12+3K	0.14	0.84	0.12	—
	12+2K	0.16	0.92	0.13	—

Table A5.73

Whilst marginally higher accuracy may be achieved with a non-linear solution there is little to recommend it over the linear version particularly when mean error and standard deviation, and extra computing overheads have been taken into account.

A5.4.2 Predicting 2 sampling intervals ahead

Table A5.74 shows the results for the linear investigation, which suggest 12 – 15 inputs for the non-linear networks. The results for these are shown below in A5.75 - 76.

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0039	0.0251	0.0039	0
4	0.0039	0.0251	0.0039	0
5	0.0038	0.0252	0.0038	0
6	0.0037	0.0249	0.0038	0
7	0.0037	0.0249	0.0038	0
8	0.0037	0.0249	0.0038	0
9	0.0037	0.0248	0.0038	0
10	0.0037	0.0237	0.0037	0
11	0.0037	0.023	0.0036	0
12	0.0037	0.0228	0.0036	1
13	0.0037	0.023	0.0036	0
14	0.0037	0.0232	0.0036	0
*15	0.0036	0.0227	0.0035	1
16	0.0036	0.023	0.0035	0
17	0.0036	0.023	0.0035	3
18	0.0036	0.023	0.0035	2
19	0.0036	0.023	0.0035	3
20	0.0036	0.023	0.0035	2

Table A5.74 – linear

inputs	cells	mean	max	std
*13	4	0.0030	0.0199	0.0029
13	5	0.0034	0.0209	0.0032
15	4	0.0034	0.0215	0.0031
*12	2	0.0037	0.0219	0.0034
15	2	0.0033	0.0220	0.0032
12	5	0.003	0.0222	0.003
13	2	0.0034	0.0222	0.0032
14	3	0.0037	0.0223	0.0034
14	2	0.0036	0.0226	0.0034
14	4	0.0031	0.0227	0.0031
15	3	0.0034	0.0228	0.0032
12	4	0.0035	0.0237	0.0032
15	5	0.0036	0.0280	0.0036
14	5	0.0054	0.0329	0.0056
13	3	0.0061	0.0355	0.0063
12	3	0.007	0.0413	0.0067

Table A5.75 – log sigmoid

inputs	cells	mean	max	std
*13	3	0.0033	0.0205	0.0030
*14	2	0.0034	0.0207	0.0031
13	4	0.0031	0.0208	0.0029
15	2	0.0034	0.0208	0.0031
14	5	0.0029	0.0212	0.0029
12	3	0.0035	0.0215	0.0033
12	4	0.0034	0.0216	0.0032
14	4	0.0032	0.0219	0.0031
15	3	0.0036	0.0222	0.0034
15	4	0.0037	0.0222	0.0033
12	2	0.0037	0.0223	0.0034
13	2	0.004	0.0235	0.0036
14	3	0.004	0.0249	0.0037
15	5	0.0032	0.0271	0.0031
12	5	0.0031	0.0278	0.003
13	5	0.0089	0.0447	0.0081

Table A5.76 – tanh sigmoid

The maximum error ranges for each pre-processing method for the linear networks are shown in A5.77. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in Table A5.78 (errors are expressed as percentages) .

function	max error range (%)
linear	2.27 – 2.51
log	1.99 – 4.13
tanh	2.05 – 4.47

Table A5.77

function	struct	mean	max	s.d.	<0.01
linear	15+K	0.36	2.27	0.35	1
log	13+4K	0.30	1.99	0.29	—
	12+2K	0.37	2.19	0.34	—
tanh	13+3K	0.33	2.05	0.30	—
	14+2K	0.34	2.07	0.31	—

Table A5.78

A5.5 Air/fuel ratio (a/f.)

The data lies within the overall range 4.11 – 4.49; therefore it was decided to experiment with division by 10 for pre-processing to bring the data level closer to that of midrange for the sigmoid activation functions, rather than division by 100.

A5.5.1 Predicting 1 sampling interval ahead (linear networks)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0011	0.0487	0.0018	0
4	0.001	0.0489	0.0018	0
5	0.001	0.0493	0.0018	0
6	0.001	0.0493	0.0019	1
7	0.001	0.0492	0.0019	0
*8	0.001	0.0491	0.0019	0
9	0.001	0.0491	0.0019	0
10	0.001	0.0492	0.0019	1
11	0.001	0.0493	0.0019	1
12	0.001	0.0494	0.0018	0
13	0.001	0.0493	0.0018	2
14	0.001	0.0493	0.0018	1
15	0.001	0.0493	0.0018	1
16	0.001	0.0493	0.0018	1
17	0.001	0.0492	0.0018	0
18	0.001	0.0494	0.0018	0
19	0.001	0.0495	0.0018	1
20	0.001	0.0495	0.0018	1

Table A5.79 – linear

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0011	0.0487	0.0018	0
4	0.001	0.0489	0.0019	0
5	0.001	0.0493	0.0018	0
6	0.001	0.0493	0.0019	1
7	0.001	0.0492	0.0019	0
*8	0.001	0.0491	0.0019	0
9	0.001	0.0491	0.0019	0
10	0.001	0.0492	0.0019	0
11	0.001	0.0493	0.0019	1
12	0.001	0.0494	0.0018	0
13	0.001	0.0493	0.0018	0
14	0.001	0.0493	0.0018	2
15	0.001	0.0493	0.0018	1
16	0.001	0.0493	0.0018	1
17	0.001	0.0493	0.0018	0
18	0.001	0.0494	0.0018	0
19	0.001	0.0495	0.0018	2
20	0.001	0.0495	0.0018	1

Table A5.80 – linear '÷10'

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0015	1.3372	0.0362	0
4	0.0015	1.337	0.0362	0
5	0.0015	1.337	0.0362	0
6	0.0015	1.337	0.0362	0
7	0.0015	1.337	0.0362	1
8	0.0015	1.337	0.0362	1
9	0.0015	1.337	0.0362	0
10	0.0015	1.3369	0.0363	0
11	0.0015	1.3371	0.0363	1
12	0.0015	1.337	0.0363	0
13	0.0015	1.337	0.0363	1
14	0.0015	1.3371	0.0363	2
15	0.0015	1.3371	0.0363	1
16	0.0015	1.3371	0.0363	1
17	0.0015	1.3371	0.0363	1
18	0.0015	1.3371	0.0364	1
19	0.0015	1.3371	0.0364	1
20	0.0015	1.3371	0.0364	2

Table A5.81 – linear 'm = 2'

inputs	*mean*	*max*	*s.d.*
*3	0.0017	0.0480	0.0021
4	0.0017	0.0481	0.0021
5	0.0017	0.0481	0.0021
6	0.0017	0.0481	0.0021
7	0.0017	0.0481	0.0021
8	0.0017	0.0481	0.0021
9	0.0017	0.0481	0.0021
10	0.0017	0.0482	0.0021
11	0.0017	0.0482	0.0021
12	0.0017	0.0481	0.0021
13	0.0017	0.0481	0.0021
14	0.0017	0.0481	0.0021
15	0.0017	0.0481	0.0021
16	0.0017	0.0481	0.0021
17	0.0017	0.0481	0.0021
18	0.0017	0.0482	0.0021
19	0.0017	0.0482	0.0021
20	0.0017	0.0482	0.0021

Table A5.81a – post-processed

Examination of the linear solutions with the exception of the 'm = 2' networks where there are extremely high maximum errors before post-processing suggests non-linear networks with some 8 – 10 inputs.

The maximum error ranges for each pre-processing method for the linear networks are shown in A5.82. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in Table A5.83 (errors are expressed as percentages) .

p.p.	*max error range (%)*
none	4.87 – 4.95
÷10	4.87 – 4.95
$m = 2$	133.71 – 133.72
	4.80 – 4.82**

** after post-processing

Table A5.82

p.p	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	3+K	0.11	4.87	0.18	0
÷ 10	3+K	0.11	4.87	0.18	0
$m = 2$	3+K	0.15	133.7	3.62	0
$m = 2$**	3+K	0.17	4.80	0.21	0

**after post-processing

Table A5.83

Of note is the high '$m = 2$' error of 134% before post-processing which reduces to less that of the other solutions after post-processing.

A5.5.2 Predicting 1 sampling interval ahead (non-linear networks)

inputs	*cells*	*mean*	*max*	*std*
*8	2	0.0137	0.0366	0.0097
9	3	0.001	0.0483	0.0018
8	4	0.001	0.0486	0.0019
9	2	0.001	0.0488	0.0018
10	5	0.001	0.0489	0.002
10	2	0.001	0.049	0.0019
8	3	0.001	0.049	0.0018
9	4	0.001	0.049	0.0018
8	5	0.001	0.049	0.0018
10	3	0.001	0.0491	0.0019
9	5	0.001	0.0492	0.0019
10	4	0.0011	0.0592	0.0023

Table A5.84 – log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*9	5	0.001	0.0481	0.0017
8	4	0.0011	0.0485	0.0021
*8	2	0.001	0.0486	0.0019
8	5	0.001	0.0486	0.0019
9	4	0.001	0.0488	0.0018
9	2	0.001	0.0490	0.0018
10	2	0.001	0.0490	0.0019
9	3	0.001	0.0490	0.0019
10	4	0.001	0.0490	0.0019
10	5	0.001	0.0490	0.0018
8	3	0.001	0.0491	0.0019
10	3	0.0011	0.0496	0.0022

Table A5.85 – tanh sigmoid

inputs	cells	mean	max	std
*10	4	0.001	0.0485	0.0019
9	4	0.001	0.0486	0.0018
*8	2	0.001	0.0487	0.0018
9	5	0.001	0.0487	0.0019
10	5	0.001	0.0487	0.0022
10	3	0.0047	0.0487	0.004
8	3	0.001	0.0488	0.0017
8	5	0.001	0.0488	0.0018
9	3	0.001	0.0489	0.0018
8	4	0.001	0.0489	0.0018
9	2	0.001	0.049	0.0018
10	2	0.001	0.049	0.0019

Table A5.86 – log sigmoid '÷10'

inputs	cells	mean	max	std
*10	2	0.0138	0.0367	0.0097
10	4	0.0022	0.0449	0.0029
8	3	0.0011	0.0484	0.002
10	5	0.0011	0.0484	0.0023
8	4	0.001	0.0487	0.0019
9	4	0.001	0.0487	0.0018
10	3	0.001	0.0488	0.0018
8	2	0.001	0.0489	0.0018
9	5	0.001	0.0489	0.0018
9	2	0.001	0.0490	0.0018
9	3	0.0011	0.0549	0.0023
8	5	0.0011	0.0942	0.0033

Table A5.87 – tanh sigmoid '÷10'

inputs	cells	mean	max	std
10	4	0.0007	0.0241	0.0009
8	3	0.0006	0.0242	0.0009
8	4	0.0007	0.0244	0.0009
9	5	0.0004	0.0246	0.0008
8	2	0.0005	0.0246	0.0008
9	2	0.0005	0.0247	0.0009
10	5	0.0005	0.0247	0.0008
9	4	0.0006	0.0248	0.0009
10	3	0.0005	0.0249	0.0008
10	2	0.0008	0.0962	0.0034
8	5	0.0006	0.1023	0.0034
9	3	0.0011	0.2111	0.0085

Table A5.88 – log sigmoid '*m* = 2'

inputs	cells	mean	max	std
*8	3	0.0018	0.0475	0.0023
10	4	0.0019	0.0475	0.0023
8	4	0.0018	0.0478	0.0023
*8	2	0.0016	0.0480	0.0022
9	5	0.0016	0.0480	0.0021
10	5	0.0016	0.0480	0.002
9	2	0.0017	0.0481	0.0021
9	4	0.0017	0.0482	0.0022
10	3	0.0016	0.0483	0.0021
10	2	0.002	0.0944	0.004
8	5	0.0018	0.1053	0.004
9	3	0.0023	0.2144	0.0088

Table A5.88a –post-processed

inputs	cells	mean	max	std
10	4	0.0007	0.0241	0.001
8	3	0.0006	0.0242	0.0009
8	2	0.0005	0.0247	0.0008
9	5	0.0005	0.0247	0.001
10	5	0.0005	0.0248	0.0008
8	4	0.0005	0.0255	0.0008
10	2	0.0007	0.0348	0.0016
10	3	0.0008	0.0400	0.0018
8	5	0.0007	0.0515	0.0018
9	4	0.0008	0.0693	0.0025
9	3	0.0009	0.0781	0.0035
9	2	0.0045	5.2734	0.143

Table A5.89 – tanh sigmoid 'm = 2'

inputs	cells	mean	Max	std
*10	3	0.0021	0.0474	0.0029
10	4	0.0019	0.0475	0.0025
8	3	0.0018	0.0476	0.0023
*10	2	0.0020	0.0480	0.0027
8	2	0.0016	0.0481	0.0022
9	5	0.0017	0.0481	0.0022
10	5	0.0016	0.0482	0.0022
8	4	0.0016	0.0489	0.0022
8	5	0.0019	0.0524	0.0028
9	4	0.002	0.0691	0.0034
9	3	0.0021	0.0810	0.0042
9	2	0.0056	5.2697	0.1430

Table A5.89a – post -processed

The maximum error ranges for each pre-processing method for the linear networks are shown in A5.90. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in table A5.91 (errors are expressed as percentages) .

p.p.	*function*	*Max error range (%)*
none	log	3.66 – 5.92
	tanh	4.81 – 4.96
÷10	log	4.85 – 4.90
	tanh	3.67 – 9.42
m = 2	log	2.41 – 21.11 (4.75 – 21.44)
	tanh	2.41 – 527.34 (4.74 – 526.97)

Table A5.90 (Post-processed data in parenthesis)

pre-proc?	*struct.*	*function*	*mean*	*max*	*s.d.*
none	8+2+K	log	1.37	3.66	0.97
	9+5+K	tanh	0.10	4.81	0.17
	8+2+K		0.10	4.86	0.19
÷10	8+2+K	log	0.10	4.85	0.19
	10+2+K	tanh	1.38	3.67	0.97
m = 2**	8+3+K	log	0.18	4.75	0.23
	6+2+K		0.16	4.80	0.22
	10+3|+K	tanh	0.21	4.74	0.29
	10+2+K		0.20	4.80	0.27

**after post-processing

Table A5.91

A5.5.3 Predicting 2 sampling intervals ahead (linear networks)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0022	0.0504	0.0030	0
4	0.0021	0.0517	0.0031	0
5	0.0021	0.0509	0.0031	0
6	0.0021	0.0528	0.0031	0
7	0.0021	0.0521	0.0031	0
8	0.0021	0.0513	0.0031	0
9	0.0021	0.0514	0.0031	0
10	0.0021	0.0521	0.0031	0
11	0.0021	0.0505	0.0031	0
*12	0.0021	0.0490	0.0029	0
13	0.0021	0.0490	0.0029	0
14	0.0021	0.0490	0.0029	0
15	0.0021	0.0491	0.0029	0
16	0.0021	0.0490	0.0029	0
17	0.002	0.0494	0.0029	0
18	0.002	0.0496	0.0029	0
19	0.002	0.0495	0.0029	0
20	0.002	0.0495	0.0029	0

Table A5.92 – linear

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0022	0.0511	0.003	0
4	0.0021	0.0530	0.0031	0
5	0.0021	0.0509	0.0031	0
6	0.0021	0.0528	0.0031	0
7	0.0021	0.0521	0.0031	0
8	0.0021	0.0513	0.0031	0
9	0.0021	0.0514	0.0031	0
10	0.0021	0.0516	0.0031	0
11	0.0021	0.0505	0.0031	0
*12	0.0021	0.0490	0.003	1
13	0.0021	0.0490	0.0029	0
14	0.0021	0.0490	0.0029	0
15	0.0021	0.0491	0.0029	0
16	0.0021	0.0490	0.0029	0
17	0.002	0.0494	0.0029	0
18	0.002	0.0495	0.0029	0
19	0.002	0.0495	0.0029	0
20	0.002	0.0495	0.0029	1

Table A5.93 – linear '÷10'

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0016	1.3369	0.0362	0
4	0.0016	1.3369	0.0362	0
5	0.0016	1.3368	0.0362	0
6	0.0016	1.3369	0.0362	0
7	0.0016	1.3369	0.0362	0
8	0.0016	1.3368	0.0362	0
9	0.0016	1.3368	0.0362	1
10	0.0016	1.337	0.0363	1
11	0.0016	1.337	0.0363	1
12	0.0016	1.3369	0.0363	1
13	0.0016	1.337	0.0363	0
14	0.0016	1.337	0.0363	0
15	0.0016	1.337	0.0363	0
16	0.0016	1.337	0.0363	0
17	0.0016	1.337	0.0364	0
18	0.0016	1.337	0.0364	0
19	0.0016	1.337	0.0364	0
20	0.0016	1.337	0.0364	1

Table A5.94 – linear '*m* = 2'

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0029	0.0468	0.0034	0
4	0.0029	0.0469	0.0034	0
5	0.0029	0.0469	0.0034	0
6	0.0029	0.0469	0.0034	0
7	0.0029	0.0468	0.0034	0
8	0.0029	0.0468	0.0034	0
9	0.0029	0.0469	0.0034	1
10	0.0029	0.0469	0.0034	1
11	0.0029	0.0469	0.0034	1
12	0.0029	0.0469	0.0034	1
13	0.0029	0.0470	0.0034	0
14	0.0029	0.0469	0.0034	0
15	0.0029	0.0470	0.0034	0
16	0.0029	0.0469	0.0034	0
17	0.0029	0.0470	0.0034	0
18	0.0029	0.0470	0.0034	0
19	0.0029	0.0470	0.0034	0
20	0.0029	0.0470	0.0034	1

Table A5.94a – linear '*m* = 2' post-processed

Examination of the linear solutions with the exception of the '$m = 2$' networks where there are extremely high maximum errors before post-processing suggests non-linear networks with some 12 – 14 inputs.

The maximum error ranges for each pre-processing method for the linear networks are shown in A5.95. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in Table A5.96 (errors are expressed as percentages) .

p.p.	*max error range (%)*
none	4.90 – 5.28
÷10	4.90 – 5.30
$m = 2$	133.68 – 133.70
	4.68 – 4.70**

** after post-processing

Table A5.95

p.p	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	12+K	0.21	4.90	0.29	0
÷ 10	12+K	0.21	4.90	0.30	1
$m = 2$	5+K	0.16	133.68	3.62	0
$m = 2$**	3+K	0.29	4.68	0.34	0

**after post-processing

Table A5.96

Again, of note is the high '$m = 2$' error of some 134% before post-processing which reduces to less that of the other matters after post-processing.

A5.5.4 Predicting 2 sampling intervals ahead (non-linear networks)

inputs	*cells*	*mean*	*max*	*std*
*12	5	0.0019	0.0474	0.0031
14	5	0.0018	0.0480	0.0028
14	3	0.0019	0.0480	0.0031
*14	2	0.0019	0.0482	0.003
13	5	0.002	0.0482	0.0035
13	2	0.002	0.0483	0.0028
13	3	0.0021	0.0487	0.0028
12	3	0.0021	0.0489	0.0031
13	4	0.0021	0.0489	0.0027
12	2	0.0021	0.0493	0.0029
12	4	0.002	0.0506	0.0035
14	4	0.003	0.1059	0.0046

Table A5.97 – log sigmoid

inputs	*cells*	*mean*	*max*	*std*
*13	2	0.0115	0.0354	0.0079
12	3	0.0020	0.0477	0.0032
13	4	0.0019	0.0478	0.0030
14	5	0.0018	0.0480	0.0031
13	3	0.0018	0.0481	0.0030
14	4	0.0018	0.0483	0.0029
12	4	0.0019	0.0486	0.0033
12	2	0.0021	0.0486	0.0029
14	2	0.0021	0.0487	0.0029
14	3	0.0021	0.0543	0.0030
13	5	0.0018	0.0849	0.0035
12	5	0.0023	0.1277	0.0051

Table A5.98 – tanh sigmoid

inputs	cells	mean	max	std
*13	3	0.0020	0.0472	0.0031
14	4	0.0021	0.0473	0.0031
12	4	0.0026	0.0476	0.0032
14	3	0.0023	0.0479	0.0031
*14	2	0.0021	0.0482	0.0028
12	3	0.0021	0.0485	0.0029
13	2	0.0021	0.0486	0.0029
12	2	0.0022	0.0487	0.0029
13	4	0.0021	0.0492	0.0030
14	5	0.0026	0.0543	0.0035
12	5	0.0022	0.2083	0.0064
13	5	0.0022	0.2150	0.0075

Table A5.99 – '÷10' log sigmoid

inputs	cells	mean	max	std
*13	4	0.0020	0.0476	0.0031
13	3	0.0019	0.0477	0.0034
12	5	0.0020	0.0477	0.0032
13	5	0.0019	0.0482	0.0034
*14	2	0.0021	0.0486	0.0029
12	4	0.0020	0.0487	0.0031
13	2	0.0021	0.0487	0.0029
12	3	0.0021	0.0487	0.0029
14	3	0.0021	0.0487	0.0028
14	5	0.0018	0.0488	0.0030
14	4	0.0020	0.0668	0.0037
12	2	0.0058	0.0712	0.0054

Table A5.100 – '÷10' tanh sigmoid

inputs	cells	mean	max	std
14	5	0.0015	1.3364	0.0363
13	4	0.0016	1.3364	0.0363
13	5	0.0015	1.3365	0.0363
13	3	0.0016	1.3366	0.0363
12	4	0.0019	1.3366	0.0366
14	2	0.0015	1.3367	0.0363
13	2	0.0017	1.3368	0.0363
12	2	0.0017	1.3369	0.0363
12	5	0.0018	1.3369	0.0364
14	4	0.0017	1.3370	0.0363
14	3	0.0018	1.3370	0.0364
12	3	0.0017	1.3371	0.0363

Table A5.101 – 'm = 2' log sigmoid

inputs	cells	mean	max	std
*14	4	0.0032	0.0464	0.0037
*12	2	0.0031	0.0465	0.0036
12	3	0.0031	0.0465	0.0037
13	4	0.0029	0.0466	0.0035
13	2	0.0030	0.0467	0.0037
14	5	0.0028	0.0469	0.0034
13	5	0.0028	0.0470	0.0034
14	2	0.0028	0.0472	0.0034
12	5	0.0032	0.0552	0.0042
14	3	0.0032	0.0601	0.0042
13	3	0.0029	0.0700	0.0039
12	4	0.0033	0.1499	0.0058

Table A5.101a – log sigmoid post-processed

inputs	cells	mean	max	std
12	3	0.0016	1.3365	0.0363
12	2	0.0015	1.3366	0.0363
13	4	0.0016	1.3366	0.0363
12	5	0.0018	1.3366	0.0368
13	5	0.0017	1.3367	0.0363
13	2	0.0016	1.3368	0.0363
14	2	0.0018	1.3368	0.0364
14	5	0.0019	1.3368	0.0368
12	4	0.0018	1.3369	0.0363
14	4	0.0017	1.3370	0.0363
14	3	0.0018	1.3370	0.0363
13	3	0.0017	1.3372	0.0363

Table A5.102 – 'm = 2' tanh sigmoid

inputs	cells	mean	max	std
*13	4	0.0030	0.0464	0.0035
12	4	0.0032	0.0465	0.0039
14	3	0.0032	0.0465	0.0038
13	3	0.0031	0.0466	0.0037
*12	2	0.0028	0.0468	0.0034
12	3	0.0029	0.0468	0.0034
13	5	0.0030	0.0468	0.0036
14	4	0.0031	0.0468	0.0037
13	2	0.0029	0.0469	0.0035
14	2	0.0032	0.0547	0.0043
14	5	0.0032	0.2097	0.0067
12	5	0.0031	0.2134	0.0071

Table A5.102a –tanh sigmoid post-processed

The maximum error ranges for each pre-processing method for the linear networks are shown in table A5.103. Solutions which demonstrate low maximum error are indicated with an asterisk and are summarised below in table A5.104 (errors are expressed as percentages) .

p.p.	*function*	*max error range (%)*
none	log	4.74– 10.59
	tanh	3.54 – 12.77
÷10	log	4.72 – 21.50
	tanh	4.76 – 7.12
m = 2	log	133.64 – 133.71 (4.64 – 14.99)
	tanh	133.65 – 133.72 (4.64 – 21.34)

Table A5.103 (Post-processed data in parenthesis)

pre-proc?	*struct.*	*function*	*mean*	*max*	*s.d.*
none	12+5+k	log	0.19	4.74	0.31
	14+2+K		0.19	4.82	0.30
	13+2+K	tanh	1.15	3.54	0.79
÷10	13+3+K	log	0.20	4.72	0.31
	14+2+K		0.21	4.82	0.28
	13+4+K	tanh	0.20	4.76	0.31
	14+2+K		0.21	4.86	0.29
m = 2	14+4+K	log**	0.32	4.64	0.37
	12+2+K		0.31	4.65	0.36
	13+4+k	tanh**	0.30	4.64	0.35
	12+2+K		0.28	4.68	0.34

**after post-processing

Table A5.104

A6 HSM Phase 2

A6.1 Predicting Wobbe Numbers

A6.1.1 Original 30-second data (1 sampling interval ahead)

inputs	mean	max	s.d.	<0.01
3	0.0022	0.0417	0.0026	0
4	0.0022	0.0415	0.0026	0
5	0.0022	0.0408	0.0026	0
6	0.0022	0.0405	0.0026	2
7	0.0022	0.0405	0.0026	1
8	0.0022	0.0424	0.0026	1
9	0.0022	0.0406	0.0026	1
10	0.0022	0.0404	0.0026	2
11	0.0022	0.0400	0.0026	1
*12	0.0022	0.0396	0.0025	2
13	0.0022	0.0398	0.0025	4
14	0.0022	0.0406	0.0025	6
15	0.0022	0.0401	0.0025	4
16	0.0022	0.0402	0.0025	6
17	0.0022	0.0402	0.0025	7
18	0.0022	0.0402	0.0025	6
19	0.0022	0.0403	0.0025	7
20	0.0022	0.0403	0.0025	8

Table A6.1 – unprocessed

inputs	mean	max	s.d.	<0.01
3	0.0022	0.0417	0.0026	0
4	0.0022	0.0415	0.0026	0
5	0.0022	0.0408	0.0026	0
6	0.0022	0.0405	0.0026	2
7	0.0022	0.0405	0.0026	1
8	0.0022	0.0424	0.0026	1
9	0.0022	0.0406	0.0026	1
10	0.0022	0.0404	0.0026	2
11	0.0022	0.0400	0.0026	1
*12	0.0022	0.0396	0.0025	2
13	0.0022	0.0398	0.0025	3
14	0.0022	0.0406	0.0025	6
15	0.0022	0.0401	0.0025	4
16	0.0022	0.0402	0.0025	6
17	0.0022	0.0402	0.0025	7
18	0.0022	0.0402	0.0025	6
19	0.0022	0.0404	0.0025	7
20	0.0022	0.0404	0.0025	8

Table A6.2 – ÷100

A6.1.2 Original 30-second data (2 sampling intervals ahead)

inputs	mean	max	s.d.	<0.01
3	0.0043	0.0859	0.0047	0
4	0.0043	0.0843	0.0047	0
5	0.0043	0.0820	0.0047	0
6	0.0043	0.0818	0.0047	0
7	0.0043	0.0816	0.0047	1
8	0.0043	0.0806	0.0047	0
9	0.0043	0.0798	0.0046	0
10	0.0043	0.0800	0.0046	0
11	0.0043	0.0789	0.0046	1
12	0.0043	0.0788	0.0046	1
13	0.0043	0.0787	0.0046	3
14	0.0043	0.0786	0.0046	3
15	0.0043	0.0786	0.0046	2
16	0.0043	0.0786	0.0046	3
17	0.0043	0.0788	0.0046	2
18	0.0043	0.0777	0.0046	3
*19	0.0043	0.0775	0.0046	4
20	0.0043	0.0776	0.0046	5

Table A6.3 – unprocessed

inputs	mean	max	s.d.	<0.01
3	0.0043	0.0860	0.0047	0
4	0.0043	0.0841	0.0047	0
5	0.0043	0.0820	0.0047	0
6	0.0043	0.0818	0.0047	0
7	0.0043	0.0816	0.0047	1
8	0.0043	0.0806	0.0047	0
9	0.0043	0.0798	0.0046	0
10	0.0043	0.0800	0.0046	0
11	0.0043	0.0789	0.0046	1
12	0.0043	0.0788	0.0046	1
13	0.0043	0.0787	0.0046	3
14	0.0043	0.0786	0.0046	2
15	0.0043	0.0786	0.0046	2
16	0.0043	0.0786	0.0046	4
17	0.0043	0.0788	0.0046	2
18	0.0043	0.0777	0.0046	3
*19	0.0043	0.0775	0.0046	4
20	0.0043	0.0776	0.0046	5

Table A6.4 – ÷100

The following tables summarise the results for both prediction intervals. When predicting two sampling intervals ahead the error ranges are approximately double those of predicting one interval ahead. The '÷100' method differs little from the unprocessed data results.

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	3.96 – 4.17	7.75 – 8.59
÷100	3.96 – 4.17	7.75 – 8.60

Table A6.5 – maximum error ranges

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	12+K	0.22	3.96	0.25	2	19+K	0.43	7.75	0.46	4
÷ 100	12+K	0.22	3.96	0.25	2	19+K	0.43	7.75	0.46	4

Table A6.6 – possible network solutions

It should be noted that in contrast to the Dawes Lane and CPS work there appear to be a significant number of potentially extraneous network inputs although the networks concerned produce the most accurate results.

A6.1.3 Synthesised 1-minute data (1 sampling interval ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0045	0.0603	0.0049	0
4	0.0045	0.0598	0.0049	0
5	0.0045	0.0597	0.0049	0
6	0.0045	0.0594	0.0049	1
*7	0.0045	0.0591	0.0048	1
8	0.0045	0.0594	0.0048	0
9	0.0045	0.0594	0.0048	0
10	0.0045	0.0594	0.0048	0
11	0.0045	0.0597	0.0048	0
12	0.0045	0.0598	0.0048	1
13	0.0045	0.0602	0.0048	0
14	0.0045	0.0604	0.0048	0
15	0.0045	0.0603	0.0048	0
16	0.0045	0.0606	0.0048	0
17	0.0045	0.0606	0.0048	1
18	0.0045	0.0606	0.0048	1
19	0.0045	0.0606	0.0048	2
20	0.0045	0.0606	0.0048	3

Table A6.7 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0045	0.0603	0.0049	0
4	0.0045	0.0598	0.0049	0
5	0.0045	0.0597	0.0049	0
6	0.0045	0.0595	0.0049	0
*7	0.0045	0.0591	0.0048	1
8	0.0045	0.0611	0.0048	0
9	0.0045	0.0606	0.0048	0
10	0.0045	0.0605	0.0048	0
11	0.0045	0.0598	0.0048	0
12	0.0045	0.0598	0.0048	1
13	0.0045	0.0603	0.0048	2
14	0.0045	0.0604	0.0048	0
15	0.0045	0.0604	0.0048	2
16	0.0045	0.0607	0.0048	0
17	0.0045	0.0606	0.0048	1
18	0.0045	0.0608	0.0048	2
19	0.0045	0.0608	0.0048	4
20	0.0045	0.0606	0.0048	3

Table A6.8 – '÷100'

A6.1.4 Synthesised 1-minute data (2 sampling intervals ahead)

inputs	*mean*	*Max*	*s.d.*	*<0.01*
3	0.007	0.1027	0.0072	0
4	0.007	0.1017	0.0072	0
5	0.0069	0.0987	0.0071	0
*6	0.0069	0.0967	0.007	1
7	0.0069	0.0971	0.007	0
8	0.0068	0.0968	0.007	0
9	0.0068	0.0968	0.0069	1
10	0.0068	0.0979	0.0069	0
11	0.0068	0.0981	0.0069	1
12	0.0068	0.0987	0.0069	0
13	0.0068	0.0991	0.0069	1
14	0.0068	0.0992	0.0069	2
15	0.0068	0.0997	0.0069	2
16	0.0068	0.0994	0.0069	2
17	0.0068	0.0994	0.0069	3
18	0.0068	0.0993	0.0069	4
19	0.0068	0.0994	0.0069	5
20	0.0068	0.0996	0.0069	4

Table A6.9 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.007	0.1028	0.0072	0
4	0.007	0.1017	0.0072	0
5	0.0069	0.0987	0.0071	0
6	0.0076	0.0994	0.0073	1
7	0.0069	0.0971	0.007	0
8	0.0068	0.0968	0.007	0
*9	0.0068	0.0964	0.0069	1
10	0.0068	0.0979	0.0069	0
11	0.0068	0.0981	0.0069	1
12	0.0068	0.0987	0.0069	0
13	0.0068	0.0991	0.0069	1
14	0.0068	0.0992	0.0069	2
15	0.0068	0.0996	0.0069	2
16	0.0068	0.0994	0.0069	2
17	0.0068	0.0994	0.0069	3
18	0.0068	0.0994	0.0069	2
19	0.0068	0.0994	0.0069	5
20	0.0068	0.0995	0.0069	5

Table A6.10 – '÷100'

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	5.91 – 6.06	9.67 – 10.27
÷100	5.91 – 6.11	9.64 – 10.28

Table A6.11 – maximum error ranges

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	7+K	0.45	5.91	0.48	1	6+K	0.69	9.67	0.70	1
÷ 100	7+K	0.45	5.91	0.48	1	9+K	0.68	9.64	0.69	1

Table A6.12 – possible network solutions

A6.2 Predicting synthesised CV

A6.2.1 30-second data (1 sampling interval ahead)

inputs	mean	max	s.d.	<0.01
*3	0.0022	0.0325	0.0024	0
4	0.0022	0.0327	0.0024	0
5	0.0022	0.0327	0.0024	1
6	0.0022	0.0327	0.0024	0
7	0.0022	0.0327	0.0024	0
8	0.0022	0.0327	0.0024	0
9	0.0022	0.0327	0.0024	1
10	0.0022	0.0327	0.0024	2
11	0.0022	0.0327	0.0024	0
12	0.0022	0.0327	0.0024	1
13	0.0022	0.0327	0.0024	2
14	0.0022	0.0327	0.0024	2
15	0.0022	0.0327	0.0024	4
16	0.0022	0.0327	0.0024	2
17	0.0022	0.0327	0.0024	2
18	0.0022	0.0327	0.0024	3
19	0.0022	0.0327	0.0024	3
20	0.0022	0.0327	0.0024	4

Table A6.13 – unprocessed

inputs	mean	max	s.d.	<0.01
*3	0.0022	0.0326	0.0024	0
4	0.0022	0.0330	0.0024	0
5	0.0022	0.0327	0.0024	1
6	0.0022	0.0328	0.0024	2
7	0.0022	0.0330	0.0024	1
8	0.0022	0.0327	0.0024	0
9	0.0022	0.0327	0.0024	1
10	0.0022	0.0327	0.0024	2
11	0.0022	0.0333	0.0024	0
12	0.0022	0.0331	0.0024	1
13	0.0022	0.0328	0.0024	2
14	0.0022	0.0330	0.0024	3
15	0.0022	0.0327	0.0024	4
16	0.0022	0.0330	0.0024	3
17	0.0022	0.0331	0.0024	2
18	0.0022	0.0327	0.0024	3
19	0.0022	0.0327	0.0024	3
20	0.0022	0.0327	0.0024	4

Table A6.14 – '÷100'

A6.2.2 30-second data (2 sampling intervals ahead)

inputs	mean	max	s.d.	<0.01
*3	0.0044	0.0721	0.0046	0
4	0.0044	0.0725	0.0046	0
5	0.0044	0.0724	0.0046	0
6	0.0044	0.0725	0.0046	2
7	0.0044	0.0725	0.0046	2
8	0.0044	0.0725	0.0046	2
9	0.0044	0.0725	0.0046	3
10	0.0044	0.0725	0.0046	2
11	0.0044	0.0727	0.0046	2
12	0.0044	0.0726	0.0046	4
13	0.0044	0.0727	0.0046	4
14	0.0044	0.0727	0.0046	4
15	0.0044	0.0727	0.0046	5
16	0.0044	0.0727	0.0046	5
17	0.0044	0.0726	0.0046	4
18	0.0044	0.0726	0.0046	3
19	0.0044	0.0727	0.0046	4
20	0.0044	0.0726	0.0046	4

Table A6.15 – unprocessed

inputs	mean	max	s.d.	<0.01
*3	0.0044	0.0723	0.0046	0
4	0.0044	0.0732	0.0046	0
5	0.0044	0.0728	0.0046	0
6	0.0044	0.0727	0.0046	1
7	0.0044	0.0728	0.0046	1
8	0.0044	0.0733	0.0046	0
9	0.0044	0.0729	0.0046	2
10	0.0044	0.0732	0.0046	0
11	0.0044	0.0732	0.0046	0
12	0.0044	0.0733	0.0046	1
13	0.0044	0.0734	0.0046	1
14	0.0044	0.0731	0.0046	1
15	0.0044	0.0727	0.0046	5
16	0.0044	0.0730	0.0046	1
17	0.0044	0.0732	0.0046	0
18	0.0044	0.0726	0.0046	4
19	0.0044	0.0736	0.0046	2
20	0.0044	0.0726	0.0046	4

Table A6.16 – '÷100'

p.p.	max error range (%)	
	1 interval ahead	2 intervals ahead
none	3.25 – 3.27	7.21 – 7.27
÷100	3.26 – 3.31	7.23 – 7.32

Table A6.17 – maximum error ranges

p.p	1 interval ahead					2 intervals ahead				
	struct	mean	max	s.d.	<0.01	struct	mean	max	s.d.	<0.01
none	3+K	0.22	3.25	0.24	0	3+K	0.44	7.21	0.46	0
÷ 100	3+K	0.22	3.26	0.24	0	3+K	0.44	7.23	0.46	0

Table A6.18 – possible network solutions

A6.2.3 1-minute data (1 sampling interval ahead)

inputs	mean	max	s.d.	<0.01
3	0.0047	0.0577	0.0049	0
4	0.0047	0.0570	0.0049	0
5	0.0047	0.0570	0.0049	0
*6	0.0047	0.0570	0.0049	0
7	0.0047	0.0572	0.0049	0
8	0.0047	0.0572	0.0049	0
9	0.0047	0.0572	0.0049	1
10	0.0047	0.0573	0.0049	0
11	0.0047	0.0573	0.0049	1
12	0.0047	0.0573	0.0049	1
13	0.0047	0.0574	0.0049	2
14	0.0047	0.0575	0.0049	3
15	0.0047	0.0574	0.0049	3
16	0.0047	0.0574	0.0049	4
17	0.0047	0.0572	0.0049	4
18	0.0047	0.0572	0.0049	5
19	0.0047	0.0570	0.0049	6
*20	0.0047	0.0569	0.0049	7

Table A6.19 – unprocessed

inputs	mean	max	s.d.	<0.01
3	0.0047	0.0577	0.0049	0
4	0.0047	0.0571	0.0049	0
*5	0.0047	0.0570	0.0049	0
6	0.0047	0.0572	0.0049	0
7	0.0047	0.0573	0.0049	0
8	0.0047	0.0573	0.0049	0
9	0.0047	0.0572	0.0049	1
10	0.0047	0.0574	0.0049	0
11	0.0047	0.0574	0.0049	2
12	0.0047	0.0575	0.0049	1
13	0.0047	0.0574	0.0049	2
14	0.0047	0.0577	0.0049	3
15	0.0047	0.0574	0.0049	3
16	0.0047	0.0576	0.0049	4
17	0.0047	0.0573	0.0049	3
18	0.0047	0.0572	0.0049	5
19	0.0047	0.0570	0.0049	6
*20	0.0047	0.0569	0.0049	7

Table A6.20 – '÷100'

A6.2.4 1-minute data (2 sampling intervals ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0074	0.1003	0.0076	0
4	0.0074	0.1003	0.0076	1
5	0.0074	0.1005	0.0076	0
6	0.0074	0.1007	0.0077	0
7	0.0073	0.1008	0.0077	1
8	0.0073	0.1008	0.0077	1
9	0.0073	0.1009	0.0077	2
10	0.0073	0.1010	0.0077	3
11	0.0073	0.1010	0.0077	4
12	0.0073	0.1011	0.0077	1
13	0.0073	0.1014	0.0077	5
14	0.0073	0.1015	0.0077	6
15	0.0073	0.1015	0.0077	4
16	0.0073	0.1014	0.0077	5
17	0.0073	0.1012	0.0077	5
18	0.0073	0.1010	0.0077	6
19	0.0073	0.1007	0.0077	6
20	0.0073	0.1006	0.0077	6

Table A6.21 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0074	0.1002	0.0076	0
4	0.0074	0.1003	0.0076	1
5	0.0074	0.1005	0.0076	0
6	0.0074	0.1007	0.0077	0
7	0.0073	0.1008	0.0077	1
8	0.0073	0.1008	0.0077	1
9	0.0073	0.1009	0.0077	2
10	0.0073	0.1010	0.0077	3
11	0.0073	0.1010	0.0077	4
12	0.0073	0.1011	0.0077	1
13	0.0073	0.1014	0.0077	5
14	0.0073	0.1015	0.0077	6
15	0.0073	0.1015	0.0077	4
16	0.0073	0.1014	0.0077	5
17	0.0073	0.1012	0.0077	5
18	0.0073	0.1010	0.0077	6
19	0.0073	0.1007	0.0077	6
20	0.0073	0.1006	0.0077	6

Table A6.22 – '÷100'

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	5.69 – 5.77	10.03 – 10.15
÷100	5.69 – 5.77	10.03 – 10.15

Table A6.23 – maximum error ranges

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	6+K	0.47	5.70	0.49	0	3+K	0.74	10.03	0.76	0
	20+K	0.47	5.69	0.49	7					
÷ 100	5+K	0.47	5.70	0.49	0	3+K	0.74	10.02	0.76	0
	20+K	0.47	5.69	0.49	7					

Table A6.24 – possible network solutions

(Note the high number of potentially extraneous inputs (7 out of 20) for the two most accurate solutions.)

A6.3 Predicting specific gravity

A6.3.1 Original 30 second data

This data lies in a range less than 1.0 and as such, requires no pre-processing.

inputs	mean	max	s.d.	<0.01
3	0.0024	0.0462	0.0031	0
*4	0.0024	0.0461	0.0032	0
5	0.0024	0.0461	0.0032	1
6	0.0024	0.0461	0.0032	0
7	0.0024	0.0462	0.0032	0
8	0.0024	0.0462	0.0032	1
9	0.0024	0.0462	0.0032	0
10	0.0024	0.0462	0.0032	1
11	0.0024	0.0462	0.0032	1
12	0.0024	0.0462	0.0032	2
13	0.0024	0.0462	0.0032	3
14	0.0024	0.0462	0.0032	2
15	0.0024	0.0462	0.0032	3
16	0.0024	0.0462	0.0032	4
17	0.0024	0.0462	0.0032	3
18	0.0024	0.0462	0.0032	3
19	0.0024	0.0462	0.0032	3
20	0.0024	0.0462	0.0032	3

Table A6.25 – 1 interval ahead

inputs	mean	max	s.d.	<0.01
3	0.0052	0.0778	0.0062	0
4	0.0051	0.0768	0.0062	0
5	0.0051	0.0767	0.0062	0
6	0.0051	0.0767	0.0062	0
7	0.0051	0.0768	0.0062	2
8	0.0051	0.0767	0.0062	1
*9	0.0051	0.0765	0.0062	2
10	0.0051	0.0766	0.0062	2
11	0.0051	0.0766	0.0062	2
12	0.0051	0.0766	0.0062	2
13	0.0051	0.0766	0.0062	3
14	0.0051	0.0766	0.0062	3
15	0.0052	0.0766	0.0062	2
16	0.0052	0.0766	0.0062	2
17	0.0052	0.0766	0.0062	4
18	0.0052	0.0766	0.0062	3
19	0.0052	0.0766	0.0062	3
20	0.0052	0.0781	0.0062	1

Table A6.26 – 2 intervals ahead

A6.3.2 Synthesised 1-minute data

inputs	mean	max	s.d.	<0.01
3	0.0056	0.0952	0.0066	0
4	0.0056	0.0950	0.0066	1
5	0.0056	0.0950	0.0066	1
6	0.0056	0.0949	0.0066	0
7	0.0056	0.0948	0.0066	0
8	0.0056	0.0947	0.0066	0
9	0.0056	0.0947	0.0067	0
10	0.0056	0.0947	0.0067	1
11	0.0056	0.0947	0.0067	2
12	0.0056	0.0949	0.0067	1
13	0.0056	0.0950	0.0066	1
*14	0.0056	0.0943	0.0067	1
15	0.0057	0.0943	0.0067	1
16	0.0057	0.0946	0.0067	0
17	0.0057	0.0947	0.0067	3
18	0.0057	0.0951	0.0067	3
19	0.0057	0.0952	0.0067	3
20	0.0057	0.0952	0.0067	3

Table A6.27 – 1 interval ahead

inputs	mean	max	s.d.	<0.01
3	0.0089	0.1561	0.0107	0
4	0.0089	0.1558	0.0107	0
5	0.0089	0.1556	0.0107	0
6	0.0089	0.1560	0.0107	0
7	0.0089	0.1557	0.0107	1
8	0.0089	0.1555	0.0107	0
9	0.0089	0.1562	0.0108	1
10	0.0089	0.1556	0.0108	1
11	0.0089	0.1557	0.0108	2
12	0.0089	0.1554	0.0108	0
13	0.0089	0.1553	0.0108	1
14	0.0090	0.1549	0.0108	2
*15	0.0090	0.1547	0.0108	1
16	0.0090	0.1554	0.0108	1
17	0.0090	0.1565	0.0109	2
18	0.0090	0.1571	0.0109	2
19	0.0090	0.1576	0.0109	2
20	0.0090	0.1572	0.0109	2

Table A6.28 – 2 intervals ahead

1 interval ahead		*2 intervals ahead*	
30-second data	*1-minute data*	*30-second data*	*1-minute data*
4.61 – 4.62	9.43 – 9.52	7.65 – 7.81	15.47 – 15.72

Table A6.29 Maximum error ranges for s.g.

intervals ahead	*30-second data*					*1-minute data*				
	Struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
1	4+K	0.24	4.61	0.32	0	14+K	0.56	9.43	0.67	1
2	9+K	0.51	7.65	0.62	2	15+K	0.90	0.1547	1.08	1

Table A6.30 – Specific gravity possible network solutions

A6.4 Predicting air/fuel ratio (a/f)

A6.4.1 Original 30 second data (1 sampling interval ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.003	0.0653	0.0038	0
4	0.003	0.0656	0.0038	0
5	0.003	0.0658	0.0037	0
6	0.003	0.0658	0.0037	1
7	0.003	0.0658	0.0037	0
8	0.003	0.0653	0.0037	0
9	0.003	0.0657	0.0037	0
10	0.003	0.0656	0.0037	0
11	0.003	0.0656	0.0037	0
12	0.003	0.0657	0.0037	1
13	0.003	0.0658	0.0037	1
14	0.003	0.0657	0.0037	3
15	0.003	0.0656	0.0037	1
16	0.003	0.0657	0.0037	3
17	0.003	0.0658	0.0037	3
18	0.003	0.0659	0.0037	3
19	0.003	0.0659	0.0037	3
20	0.003	0.0659	0.0037	4

Table A6.31 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.003	0.0653	0.0038	0
4	0.003	0.0656	0.0038	0
5	0.003	0.0658	0.0037	0
6	0.003	0.0658	0.0037	1
7	0.003	0.0658	0.0037	0
8	0.003	0.0653	0.0037	0
9	0.003	0.0657	0.0037	0
10	0.003	0.0656	0.0037	0
11	0.003	0.0656	0.0037	0
12	0.003	0.0657	0.0037	1
13	0.003	0.0657	0.0037	1
14	0.003	0.0657	0.0037	2
15	0.003	0.0656	0.0037	1
16	0.003	0.0658	0.0037	1
17	0.003	0.0658	0.0037	3
18	0.003	0.0659	0.0037	2
19	0.003	0.0659	0.0037	3
20	0.003	0.0659	0.0037	4

Table A6.32 – '÷100'

A6.4.2 Original 30 second data (2 sampling intervals ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0062	0.1220	0.0073	0
4	0.0061	0.1227	0.0072	0
5	0.0061	0.1228	0.0072	0
6	0.0061	0.1227	0.0072	0
7	0.0061	0.1228	0.0071	1
8	0.0061	0.1230	0.0071	0
9	0.0061	0.1232	0.0071	0
10	0.0061	0.1234	0.0071	0
11	0.0061	0.1236	0.0071	1
12	0.0061	0.1236	0.0071	0
13	0.0061	0.1236	0.0071	2
14	0.0061	0.1236	0.0071	0
15	0.0061	0.1236	0.0071	2
16	0.0061	0.1236	0.0071	1
17	0.0061	0.1237	0.0071	2
18	0.0061	0.1236	0.0071	2
19	0.0061	0.1236	0.0071	3
20	0.0061	0.1235	0.0071	3

Table A6.33 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
*3	0.0062	0.1220	0.0073	0
4	0.0061	0.1227	0.0072	0
5	0.0061	0.1228	0.0072	0
6	0.0061	0.1227	0.0072	0
7	0.0061	0.1228	0.0071	1
8	0.0061	0.1230	0.0071	0
9	0.0061	0.1232	0.0071	0
10	0.0061	0.1234	0.0071	0
11	0.0061	0.1235	0.0071	0
12	0.0061	0.1236	0.0071	0
13	0.0061	0.1235	0.0071	1
14	0.0061	0.1236	0.0071	0
15	0.0061	0.1236	0.0071	2
16	0.0061	0.1236	0.0071	1
17	0.0061	0.1236	0.0071	2
18	0.0061	0.1236	0.0071	2
19	0.0061	0.1236	0.0071	2
20	0.0061	0.1235	0.0071	3

Table A6.34 – '÷100'

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	6.53 – 6.59	12.20 – 12.36
÷100	6.53 – 6.59	12.20 – 12.36

Table A6.35 – maximum error ranges, a/f 30 second data

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	3+K	0.30	6.53	0.38	0	3+K	0.62	12.20	0.73	0
÷100	3+K	0.30	6.53	0.38	0	3+K	0.62	12.20	0.73	0

Table A6.36 – possible network solutions, a/f 30 second data

A6.4.3 1-minute data (1 sampling interval ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0059	0.1160	0.0062	0
4	0.0059	0.1167	0.0062	0
5	0.0059	0.1149	0.0062	0
6	0.0059	0.1139	0.0061	0
7	0.0059	0.1137	0.0061	0
*8	0.0059	0.1136	0.0061	1
9	0.0059	0.1137	0.0061	1
10	0.0059	0.1136	0.0061	2
11	0.0059	0.1136	0.0061	3
12	0.0059	0.1135	0.0061	3
13	0.0059	0.1137	0.0061	4
14	0.0059	0.1137	0.0062	3
15	0.0059	0.1144	0.0062	5
16	0.0059	0.1143	0.0062	6
17	0.0059	0.1142	0.0062	4
18	0.0059	0.1143	0.0062	5
19	0.0059	0.1141	0.0062	5
20	0.0059	0.1142	0.0062	5

Table A6.37 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0059	0.1161	0.0062	0
4	0.0059	0.1187	0.0062	1
5	0.0059	0.1195	0.0062	0
6	0.0059	0.1182	0.0062	0
*7	0.0059	0.1136	0.0061	0
8	0.0059	0.1138	0.0061	1
9	0.0059	0.1136	0.0061	1
10	0.0059	0.1136	0.0061	2
11	0.0058	0.1191	0.0062	0
12	0.0059	0.1144	0.0061	1
13	0.0059	0.1138	0.0061	4
14	0.0059	0.1137	0.0062	3
15	0.0059	0.1145	0.0062	5
16	0.0059	0.1145	0.0062	6
17	0.0059	0.1143	0.0062	4
18	0.0059	0.1143	0.0062	5
19	0.0059	0.1140	0.0062	4
20	0.0059	0.1142	0.0062	4

Table A6.38 – '÷100'

A6.4.4 1-minute data (2 sampling intervals ahead)

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0094	0.1456	0.0095	0
4	0.0094	0.1446	0.0095	0
5	0.0094	0.1414	0.0095	0
6	0.0093	0.1406	0.0095	1
*7	0.0092	0.1405	0.0095	0
8	0.0092	0.1405	0.0095	0
9	0.0092	0.1406	0.0095	1
10	0.0092	0.1406	0.0095	1
11	0.0092	0.1406	0.0095	1
12	0.0092	0.1407	0.0095	1
13	0.0092	0.1409	0.0095	2
14	0.0093	0.1419	0.0095	1
15	0.0093	0.1424	0.0095	2
16	0.0093	0.1422	0.0095	2
17	0.0093	0.1423	0.0095	3
18	0.0093	0.1421	0.0095	4
19	0.0093	0.1418	0.0095	6
20	0.0093	0.1418	0.0095	7

Table A6.39 – unprocessed

inputs	*mean*	*max*	*s.d.*	*<0.01*
3	0.0094	0.1456	0.0095	0
4	0.0094	0.1446	0.0095	0
5	0.0094	0.1413	0.0095	0
6	0.0093	0.1406	0.0095	1
7	0.0092	0.1405	0.0095	0
8	0.0092	0.1404	0.0095	0
9	0.0092	0.1406	0.0095	1
*10	0.0092	0.1364	0.0095	3
11	0.0092	0.1406	0.0095	1
12	0.0092	0.1408	0.0095	1
13	0.0092	0.1409	0.0095	1
14	0.0093	0.1418	0.0095	1
15	0.0093	0.1424	0.0095	2
16	0.0093	0.1423	0.0095	2
17	0.0093	0.1423	0.0095	3
18	0.0093	0.1421	0.0095	3
19	0.0093	0.1419	0.0095	6
20	0.0093	0.1418	0.0095	7

Table A6.40 – '÷100'

p.p.	*max error range (%)*	
	1 interval ahead	*2 intervals ahead*
none	11.36 – 11.67	14.05 – 14.56
÷100	11.36 – 11.91	13.64 – 14.56

Table A6.41 – maximum error ranges, a/f 1minute data

p.p	*1 interval ahead*					*2 intervals ahead*				
	struct	*mean*	*max*	*s.d.*	*<0.01*	*struct*	*mean*	*max*	*s.d.*	*<0.01*
none	8+K	0.59	11.36	0.61	1	7+K	0.92	14.05	0.95	0
÷100	7+K	0.59	11.36	0.61	1	10+K	0.92	13.64	0.95	3

Table A6.42 – possible network solutions, a/f 1minute data

A.7 Microprocessor-based Systems and Hardware Limitations

This appendix outlines the basic differences between PLCs (Progammable Logic Controllers) and general purpose computers. It is not intended as an in-depth study but to serve as an informal introduction to the topic for non-IT specialist readers. For more detailed treatments the reader is referred to specialist texts [e.g. Mano88, Kiss86]. An effective overview of PLC ladder logic programming and other related issues, which the author has used in the course of teaching undergraduates may be found at [PLCS05]. Details of pricing and datasheets for individual component types may be obtained from major UK suppliers such as Farnell, Maplin, RS Components. (Contact details are given in the main reference section.)

A.7.1 Microprocessor-based systems

Microprocessors are single-chip devices that together with a few additional components (e.g. memory chips, chips interfacing to external equipment such as keyboards and monitors, printers, etc) are capable of forming the bases of a complete computer.

The microprocessor itself consists primarily of an ALU (Arithmetic and Logic Unit) surrounded by temporary (data) storage registers. The ALU is capable of performing Boolean logic operations on the contents of these registers and in its most basic form, elementary arithmetic operations (add, subtract, multiply and divide) on integer values. Because the registers are integral to the chip and access the ALU directly, register operations are extremely fast (at the speed of light).

In the case of a basic integer-arithmetic ALU, operations involving floating point values require additional software programs located externally to the chip in system memory. This involves numerous transfers via the data bus between memory and the microprocessor registers. As a result the speed of operation is limited by the speed of the bus which is dependent on the system clock (a signal which synchronises the microprocessor's actions), the width of the data bus (in bits), and the number of transfers required.

E.g. An IEEE-standard 32-bit representation of a floating point value on an 8-bit bus would require 4 transfer operations, whereas a 16-bit bus would require only 2.

The time taken for each individual transfer is dependent on system clock speed which in turn is limited by the physical characteristics of the system components. The faster the operational speed of the component, the higher the cost.

The program requirement for data transfers to and from external memory is decreased as number of data storage registers on the processor is increased. Again component (microprocessor) cost is a factor.

More recently floating point operations have been performed in hardware as facilities for these together with more sophisticated functions such as trigonometric and exponential calculations have been incorporated into microprocessor hardware. However there is a substantial cost differential between the simpler processors and these more sophisticated examples. For example the MC6800 processor which formed the basis of many of the older PLC types was less than 10% of the cost of an MC68000 processor used in the earlier Apple computers.

A.7.2 Limitations of Programamble Logic Controllers (PLCs)

A PLC a specific implementation of a microprocessor system whose primary purpose is to emulate the behaviour of an industrial electrical switching system through the use of virtual relays, timers and counters. The arrangement and interconnection of these virtual devices is dictated by a user program most conventionally written using 'ladder logic' where the various switching circuits appear to be represented by the rungs of a ladder [e.g. PLCS05]. This program is in turn, executed largely as a series of Boolean logic statements with some integer arithmetic operations.

Thus a sophisticated ALU with complex floating point capabilities is not normally required and manufacturing costs for the device can be reduced correspondingly. Further, the results of these operations are largely Boolean 'true' or 'false' which can be represented as a single 'bit' rather than a series of '8-bit' bytes. (A 32-bit floating point number would require the memory space of 32 such Boolean results above.) Hence further cost reductions are possible with respect to memory requirements. In addition, the operational speeds of the electro-mechanical systems being replaced by PLCs are relatively low; thus a low-speed, low-cost microprocessor is adequate for the majority of implementations, again with potential for cost reduction.

In summary, the reduced memory capacity places severe restrictions on the amount of data storage space and length of program. Further, fewer registers and reduced bus size and speed constrain the number of arithmetic calculations that may be performed with in a given time interval. Thus for a neural network implementation to attain a reasonable operational speed it is advantageous to minimise the number of computations, and memory requirements, by

minimising the number of network inputs and by utilising linear activation functions rather than the more computationally demanding non-linear functions.